

# EatCom

Red social temática.



Grado en Ingeniería Multimedia

## Trabajo Fin de Grado

Autor:  
Francisco Manuel Gómez Pastor

Tutor/es:  
José Vicente Berná Martínez

Septiembre 2018



Universitat d'Alacant  
Universidad de Alicante



# Resumen

En el presente Proyecto de Fin de Grado se ha desarrollado una aplicación de una red social vertical cuya temática es la restauración. El sistema está compuesto por una aplicación móvil y un backoffice para la moderación del contenido por los administradores.

Hoy en día el teléfono móvil se ha vuelto una herramienta indispensable de uso diario, que consultamos para ver qué tiempo va a hacer, leer las últimas noticias, estar al tanto de las últimas tendencias en Instagram o incluso buscar lugares para comer o cenar.

Hace bastantes años que soy miembro del grupo de Facebook “Comer BBB (Bien, Bueno y Barato) - Benidorm y Provincia” en el cual la gente comparte sus experiencias de los restaurantes a los que va. Hay veces que entro al grupo para encontrar algún sitio nuevo al que ir o simplemente para ver qué locales están de moda. Aunque cuando estoy de vacaciones esta opción no me vale, porque estos grupos son cerrados y abarcan solo una ciudad o provincia. Es por eso por lo que utilizo TripAdvisor o Google Maps.

Por estas razones surge la idea de crear EatCom; desarrollar una aplicación que junte la parte social de Facebook con el sistema de valoración y los filtros de las aplicaciones como TripAdvisor.

En la memoria de este proyecto se ha realizado un estudio de las diferentes tecnologías existentes y las que mejor se adaptaban al proyecto, dando como resultado un proyecto basado en el MEAN stack, usando un API Rest para el backend y utilizando HTTP y JSON para la transmisión de información entre el frontend y el backend.

El proyecto se ha desarrollado siguiendo la metodología en cascada, definiendo objetivos, analizando los requisitos funcionales y no funcionales, definiendo el diseño de interfaces y el diseño del API Rest.

Como resultados, se ha obtenido un sistema completo y funcional que cumple con las especificaciones definidas.

## Motivación, justificación y objetivo general

Durante el cuarto curso de carrera trabajamos con la metodología ABP (Aprendizaje Basado en Proyectos) con la que tuvimos que plantear, investigar, diseñar y llevar a cabo un proyecto desde cero.

Recuerdo que el primer día no teníamos ni idea de que tecnología usar, era un reto que no se nos había planteado hasta ahora, ya que durante la carrera había sido todo un camino guiado del cual no nos podíamos salir. El ABP nos daba la oportunidad de crear algo desde cero con las tecnologías que quisiéramos, por lo que miramos cual era la tendencia y usamos aquellas que no habíamos visto hasta ahora y nos parecían más atractivas. Al final el resultado fue bastante bueno y nos ayudó a cada integrante del grupo a ver en que queríamos especializarnos de cara al futuro, que en mi caso fue el desarrollo tanto de *frontend* como de *backend* o lo que llamaríamos un desarrollador *fullstack*.

Después de terminar el ABP y gracias a que había tocado Node.js durante el proyecto, conseguí hacer unas prácticas en una empresa en la cual pude seguir desarrollándome, con el mantenimiento y creación de servicios de comunicación entre almacenes y la integración de APIs externas en el sistema.

Llegado este punto y siendo este mi último año de carrera, quiero afrontar el trabajo fin de grado como un reto, donde poner en práctica los conocimientos adquiridos durante estos años, tener la oportunidad de formarme en nuevas tecnologías y perfeccionar las que ya conozco.

Desde hace unos años pertenezco a un grupo de Facebook llamado “Comer BBB (Bien, Bueno y Barato) - Benidorm y Provincia”. En este grupo la gente pone comentarios de los restaurantes a los que va, hablando de su experiencia y adjuntando fotos de los platos. También tenemos otra gente que pide recomendaciones sobre sitios a los que ir ya sea por tipo de comida, precio o zona.

En mi caso, suelo seguir este grupo para descubrir nuevos sitios, ya que en webs como Tripadvisor o Google Maps, para ver los comentarios de la gente tengo que buscar un restaurante en concreto, por lo que primero suelo entrar al grupo de Facebook para ver las últimas recomendaciones de la gente y luego busco más información.

Así es como surge mi idea, juntar la posibilidad de ver los comentarios de la gente sobre los sitios a los que ha ido y a la vez tener la información de los restaurantes es un mismo sitio. De esta

forma cuando queramos conocer algún sitio nuevo, tan solo tendremos que entrar y ver las recomendaciones de la gente.

Mi proyecto estará basado en una red social vertical o temática, que girará en torno a los restaurantes y la opinión de los usuarios sobre estos.

## Agradecimientos

Agradecer a mi familia todo el apoyo obtenido desde el primer día que decidí hacer la carrera de Ingeniería Multimedia.

No puedo dejar fuera a José Vicente Berná, que me ha guiado en este proyecto dándome los consejos que necesitaba y mostrándome otras perspectivas.

Por último, pero sin duda no menos importante, quiero agradecer a Anna Mestre el haber estado ahí en todo momento, siendo pesada a su manera y animándome. Ha sido una parte muy importante en esta última etapa y sin ella no hubiera sido posible.

## Citas

“En la vida todo llega, todo pasa y todo cambia.”

# Índice de contenidos

Resumen.....	2
Motivación, justificación y objetivo general.....	3
Agradecimientos .....	5
Citas.....	6
Índice de figuras .....	10
Índice de tablas .....	12
1. Introducción.....	13
2. Estado del arte .....	15
2.1. Aplicaciones centradas en la opinión del usuario .....	15
2.1.1. Google .....	16
2.1.2. Tripadvisor.....	18
2.1.3. Yelp.....	21
2.1.4. Facebook .....	23
2.2. Aplicaciones para realizar reservas .....	24
2.2.1. ElTenedor .....	25
2.3. Aplicaciones para pedir comida a domicilio .....	27
2.3.1. Just Eat .....	27
3. Objetivos .....	30
4. Metodología.....	31
5. Análisis y especificación .....	33
5.1. Descripción general .....	33
5.1.1. Perspectiva del producto .....	33
5.1.2. Funciones del producto.....	33
5.1.3. Características de los usuarios .....	33
5.1.4. Restricciones .....	34
5.1.5. Suposiciones y dependencias.....	34



5.2.	Especificación y requisitos .....	34
5.2.1.	Requisitos funcionales.....	34
5.2.2.	Requisitos no funcionales .....	38
5.2.3.	Requisitos deseables .....	39
6.	Diseño .....	41
6.1.	Diseño de la persistencia .....	41
6.2.	Diseño arquitectura conceptual .....	42
6.3.	Diseño API Rest.....	43
6.3.1.	Rutas de comentario .....	44
6.3.2.	Rutas de imágenes .....	46
6.3.3.	Rutas de usuario .....	47
6.3.4.	Identificación mediante Google OAuth API .....	48
6.3.5.	Google Places API Web Service.....	49
6.4.	Diseño arquitectura tecnológica Front/Backend .....	50
6.4.1.	Frontend.....	50
6.4.2.	Backend .....	51
6.5.	Diseño de la Experiencia de Usuario .....	51
6.6.	Diseño Interfaces .....	53
7.	Implementación .....	59
7.1.	Entorno de desarrollo .....	59
7.2.	Implementación Backend .....	59
7.2.1.	Instalación de dependencias.....	59
7.2.2.	APIs de Google .....	60
7.2.2.1.	Geolocation.....	60
7.2.2.2.	Place Search .....	60
7.2.2.3.	Place Details.....	61
7.2.3.	Controladores.....	61
7.3.	Implementación Frontend .....	61

7.4.	Producto final .....	62
8.	Resultados .....	66
9.	Conclusiones y trabajo futuro .....	68
9.1.	Mejoras y ampliaciones.....	68
9.2.	Conclusión .....	68
	Bibliografía .....	69

# Índice de figuras

Figura 1. Google - Restaurantes de sushi.....	16
Figura 2. Google - Versión móvil .....	17
Figura 3. Tripadvisor - Información de un restaurante.....	18
Figura 4. Tripadvisor – Opiniones .....	19
Figura 5. Tripadvisor - Escribir una opinión .....	20
Figura 6. Tripadvisor - Escribir opinión desde app.....	20
Figura 7. Yelp - Ficha de un restaurante .....	21
Figura 8. Yelp - Error en el nombre de un restaurante .....	22
Figura 9- Yelp - Escribir reseña.....	22
Figura 10. Facebook - Escribir opinión sobre un restaurante.....	23
Figura 11. Facebook - Reseña de un usuario .....	24
Figura 12. ElTenedor – Buscador .....	25
Figura 13. ElTenedor - Ficha de un restaurante.....	25
Figura 14. ElTenedor – Opiniones .....	26
Figura 15. ElTenedor - Carta del restaurante.....	26
Figura 16. Just Eat - Resultado de la búsqueda .....	27
Figura 17. Just Eat - Ficha de un restaurante.....	28
Figura 18. Proceso de desarrollo Scrum .....	31
Figura 19. Etapas del modelo en cascada .....	32
Figura 20. Modelo de datos .....	42
Figura 21. Esquema conceptual de la arquitectura .....	43
Figura 22. Arquitectura del servidor .....	44
Figura 23. Flujo para identificar un usuario .....	49
Figura 24. Ejemplo de cambio de menú .....	52
Figura 25. Inicio mostrando reseñas y restaurantes.....	54
Figura 26. Buscador con resultados.....	54
Figura 27. Localización .....	55
Figura 28. Ficha de un restaurante .....	55
Figura 29. Identificación de usuario.....	56
Figura 30. Escribir una reseña .....	57
Figura 31. Perfil personal .....	57

Figura 32. Editar perfil personal.....	58
Figura 33. Vista de administrador .....	58
Figura 34. Página principal – Comentarios .....	62
Figura 35. Página principal - Restaurantes.....	62
Figura 36. Nueva crónica sin identificar usuario.....	63
Figura 37. Nueva crónica con usuario identificado.....	63
Figura 38. Formulario para crear una crónica.....	64
Figura 39. Perfil de usuario .....	64
Figura 40. Datos personales.....	65

## Índice de tablas

Tabla 1. Webs analizadas .....	14
--------------------------------	----

# 1. Introducción

Los seres humanos, al igual que el resto de los seres vivos, necesitamos una variada y equilibrada alimentación para la vida.

Entendemos como alimentación la ingesta de alimento por parte de los organismos para proveerse de sus necesidades alimenticias, fundamentalmente para conseguir energía y desarrollarse.

La comida es la fuente que nos proporciona los medios necesarios para esto, pero además de ser una necesidad, la comida es un placer.

En más de una ocasión he quedado para comer o cenar con amigos y no hemos sabido a que sitio ir y hemos acabado yendo al mismo restaurante aburrido de siempre. Es un problema recurrente que al igual que yo, tienen muchas personas.

Normalmente cuando salimos a comer fuera de casa solemos ir a sitios de confianza o que nos han recomendado, pero cuando ya nos hemos cansado de ir a los mismos sitios, o sobre todo cuando estamos de turismo o visitando otras ciudades, tenemos que averiguar cuáles son los mejores restaurantes.

Existen páginas o aplicaciones muy potentes en el mercado como Tripadvisor, ElTenedor o Google que nos ayudan en la tarea de encontrar nuevos restaurantes a los que ir, dándonos diferentes maneras de filtrar los resultados para encontrar el restaurante perfecto, pero en todos ellos, si queremos conocer la opinión de los usuarios, primero debemos entrar a la ficha de un restaurante y después leer las opiniones sobre este.

Por otro lado, están los grupos de Facebook con temática de restauración. En estos grupos hay usuarios que piden recomendaciones sobre restaurantes para que el resto de los usuarios les aconsejen y otros usuarios que comparten su experiencia en los restaurantes a los que han ido, describiendo que comieron, como fue el trato, precio y adjuntando imágenes de los platos y el local. Este concepto me gusta mucho ya que suelo entrar para ver donde ha ido la gente últimamente o que lugares están de moda y a veces descubro sitios que, de no ser por el grupo, quizás no habría llegado a conocer.

La diferencia entre las web o aplicaciones comentadas con respecto a los grupos de Facebook es que las primeras están orientadas al negocio, es decir, se centran en ofrecer servicio al restaurante y es ahí por donde empieza la búsqueda, primero buscas sitios y luego dentro de

cada uno puedes ver los comentarios. Sin embargo, los grupos se centran en los usuarios, en sus opiniones y lo que aportan, lo malo es que las opiniones son temporales, es decir, uno acaba viendo las últimas opiniones y es difícil localizar por otros factores como el lugar, la media de las opiniones, etc.

El principal problema que tienen los grupos es que están segregados por regiones, es decir, abarcan una ciudad y alrededores o una provincia, por lo que lo más normal es pertenecer al grupo de tu ciudad de residencia. Además, la mayoría de los grupos de este estilo suelen ser cerrados y requieren de un administrador para que acepte tu solicitud, de manera que la admisión no es instantánea, lo que es un inconveniente cuando estamos de vacaciones y no sabemos dónde ir.

Este proyecto pretende juntar estos dos conceptos en uno, para que el usuario que está de vacaciones pueda ver de un vistazo cientos de comentarios y recomendaciones sobre distintos restaurantes cercanos, o ese usuario que ya se conoce todos los restaurantes de su ciudad y busca probar algo nuevo, entre en la plataforma y de manera inesperada vea una recomendación de ese local discreto del que no se había percatado hasta ahora y que de hecho tiene muy buena recomendación.

EatCom busca crear esa sensación de serendipia en los usuarios cuando entren, que tengan la posibilidad de encontrar su nuevo local favorito de forma casual además de compartir sus experiencias, consultar información detallada de los restaurantes, filtrar los resultados para una mejor búsqueda, etc.

## 2. Estado del arte

Existen distintas webs o aplicaciones que ofrecen algunas de las funcionalidades que se pretenden desarrollar en este proyecto, pero ninguna con el enfoque que se le pretende dar. Sin embargo, es útil analizar las alternativas que existen actualmente para determinar las prioridades a la hora de desarrollar las funcionalidades y proporcionarnos una visión global del problema y las diferentes soluciones.

A pesar de que todas las webs disponen de opiniones de usuario, para el análisis las he agrupado en tres categorías según el servicio que ofrecen. En el primer grupo es el más numeroso y estarían aquellas que están centradas en la opinión del usuario; en el segundo grupo webs para realizar reservas; y en el tercero, webs para pedir comida a domicilio.

Todas las webs que se van a analizar disponen de aplicación, por lo que primero analizaremos la web y después veremos su propuesta en aplicación. Para todas las web y aplicaciones usaremos la zona de Benidorm para obtener los restaurantes cercanos.

*Tabla 1. Webs analizadas*

Logo	Nombre	Url
	Google	<a href="https://www.google.es">https://www.google.es</a>
	Tripadvisor	<a href="https://www.tripadvisor.es/">https://www.tripadvisor.es/</a>
	Yelp	<a href="https://www.yelp.es">https://www.yelp.es</a>
	Facebook	<a href="https://www.facebook.com/">https://www.facebook.com/</a>
	ElTenedor	<a href="https://www.eltenedor.es/">https://www.eltenedor.es/</a>
	Just Eat	<a href="https://www.just-eat.es/">https://www.just-eat.es/</a>

### 2.1. Aplicaciones centradas en la opinión del usuario

En este tipo de aplicaciones, su principal servicio es ofrecer cientos de valoraciones de usuarios sobre los distintos restaurantes.



### 2.1.1. Google

Voy a empezar analizando la que para mí sería la principal herramienta de búsqueda para cualquier cosa, incluida la búsqueda de un restaurante.

Con Google podemos buscar un restaurante en concreto o realizar búsquedas del tipo “restaurantes cercanos”, “restaurante italiano”, “restaurantes baratos” y nos ofrecerá una lista con los restaurantes cercanos que encajen con la búsqueda.

Para filtrar el contenido tenemos dos opciones que se pueden complementar, por un lado, podemos seleccionar la valoración mínima que queremos que tengan los resultados que nos muestre y por otro el horario de apertura, en el que podemos indicar si queremos un día y hora concreta, que estén abiertos a la hora que se realiza la búsqueda, abierto 24 horas o cualquier hora.

En la Figura 1 podemos ver un ejemplo de búsqueda y la manera que tiene Google de representar los resultados.

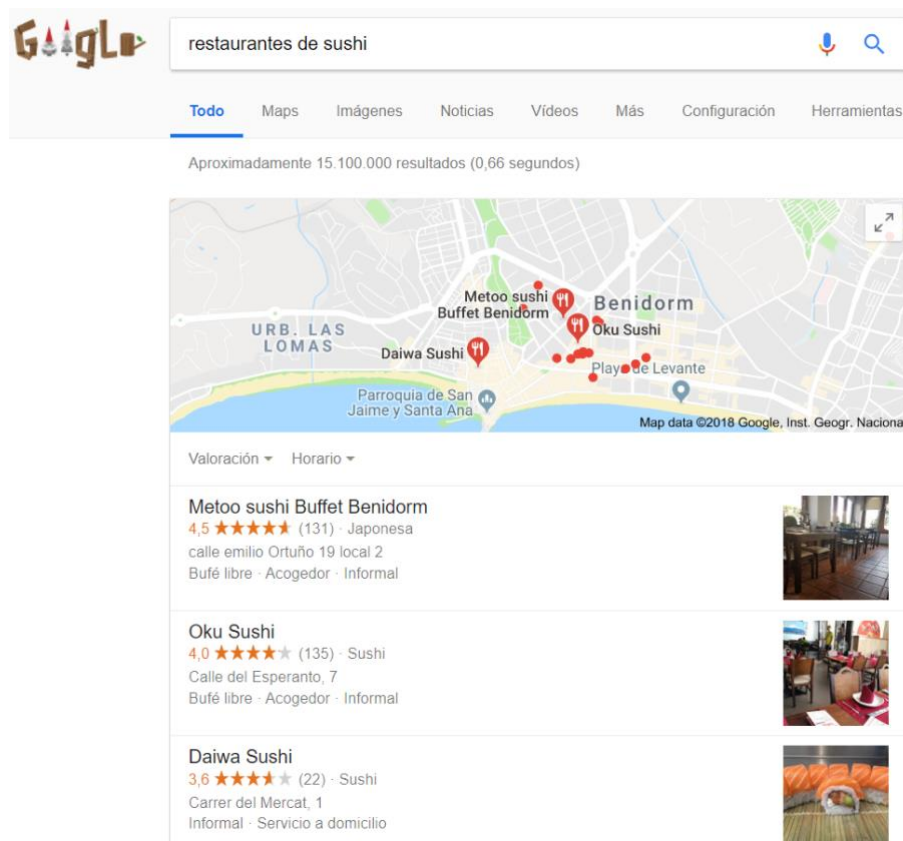


Figura 1. Google - Restaurantes de sushi  
(Fuente <https://www.google.es>)

Al hacer clic en uno de los resultados nos mostrará su posición en el mapa, toda la información del restaurante como la dirección, horario de apertura y teléfono. También podemos ver preguntas y respuestas de los usuarios, los horarios populares y el tiempo medio que suele pasar la gente en ese local, fotos y valoraciones de los usuarios.

Al escribir una reseña nos aparecerá una ventana en la que podremos puntuar mediante estrellas del 1 al 5, escribir los detalles de la experiencia en el lugar y adjuntar imágenes.

La versión móvil de Google nos ofrece exactamente la misma información que la web y estructurada de la misma manera cómo podemos apreciar en la Figura 2. En este caso se ha adaptado la información web a una versión móvil.

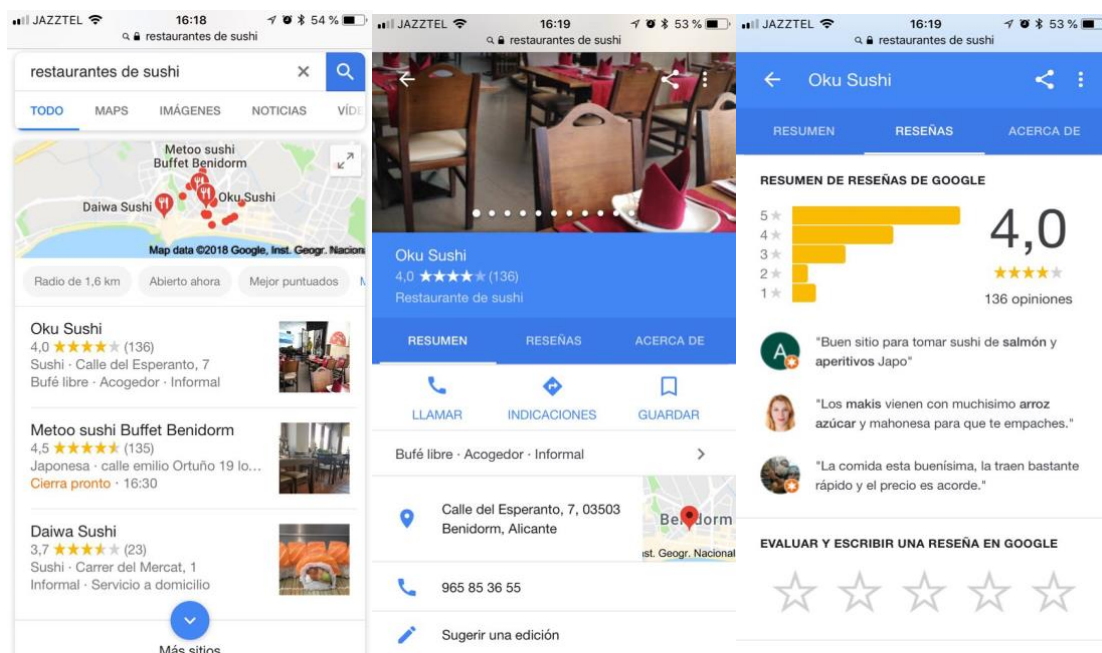


Figura 2. Google - Versión móvil  
(Fuente <https://www.google.es>)

Destacaré de Google la forma de mostrar los resultados, ya que de un vistazo podemos ver tanto en forma de listado como gráficamente en el mapa los principales resultados. Es muy intuitivo de usar y tiene toda la información necesaria sobre el restaurante.

Para filtrar los resultados solo disponemos de un filtro sobre el horario de apertura del local y otro para la puntuación lo que limita un poco a la hora de querer acotar por tipos de cocina, etc. Pero esto lo suple con su buscador.

### 2.1.2. Tripadvisor

TripAdvisor no está únicamente especializado en restauración, pero tiene una sección muy importante dedicada a ello.

Al entrar ya nos ofrece una variedad de sitios según la ciudad que hayamos buscado la última vez, agrupados por diferentes criterios.

Al buscar por ciudad nos ofrecerá una lista inmensa de restaurantes divididos en diferentes grupos como restaurantes elegantes, de precio moderado, cocina local etc. Para ayudarnos en la búsqueda podemos elegir entre numerosos filtros para encontrar el restaurante ideal, como son el tipo de cocina, tipo de comida, rango de precios, etc.

En la ficha de los restaurantes encontraremos información detallada del restaurante, fotos y comentarios de los usuarios y su ubicación.

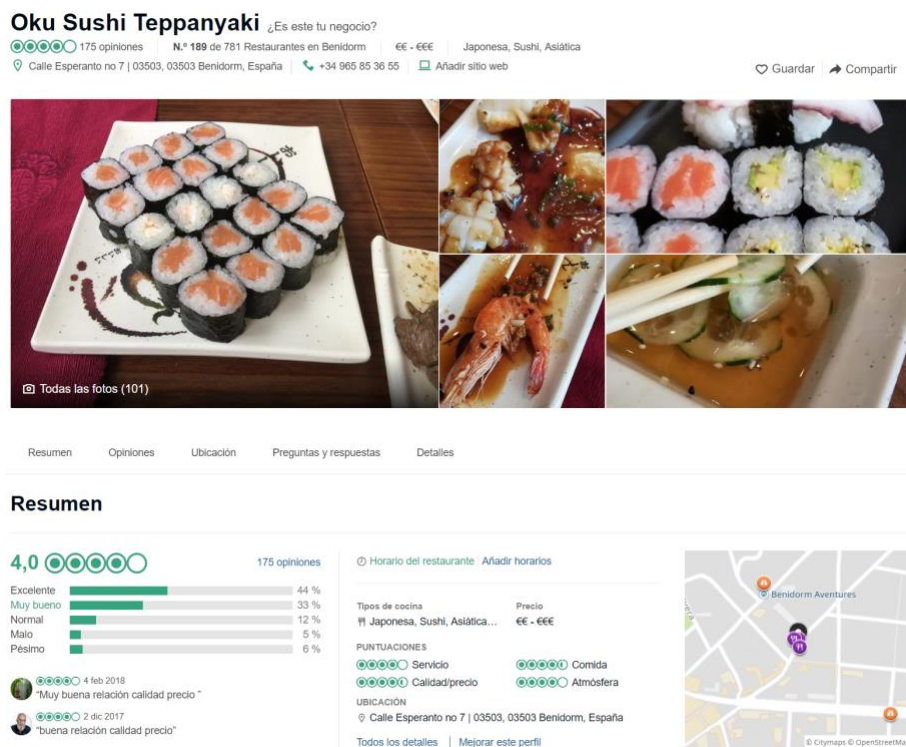


Figura 3. Tripadvisor - Información de un restaurante  
(Fuente <https://www.tripadvisor.es>)

Al igual que con los resultados de los restaurantes, también dispone de varios filtros para segmentar las opiniones de los usuarios como podemos apreciar en la Figura 4. Pudiendo no solo filtrar por puntuación, sino por el tipo de viajero, la época del año en la que fue al restaurante o el idioma. Incluso dispone de un buscador de palabras clave para las opiniones.

## Opiniones (175)

Escribe una opinión

### Puntuación de viajeros



### Tipo de viajero

- ☐ Familias
- ☐ En pareja
- ☐ En solitario
- ☐ De negocios
- ☐ Amigos

### Época del año

- ☐ Mar-may
- ☐ Jun-ago
- ☐ Sep-nov
- ☐ Dic-feb

### Idioma

- ☐ Todos los idiomas
  - ☒ Español (148)
  - ☐ Inglés (39)
  - ☐ Holandés (9)
- [Más idiomas](#)

### Lee lo que dicen los viajeros:

Buscar opiniones


1 - 10 de 148 opiniones

Figura 4. Tripadvisor – Opiniones  
(Fuente <https://www.tripadvisor.es>)

Entre la versión web y la aplicación móvil las principales diferencias que se pueden apreciar es que la aplicación móvil es algo más sencilla, sobre todo a la hora de escribir una opinión. En la Figura 5 podemos ver el proceso completo para escribir una opinión desde la web y en la Figura 6 el de la aplicación móvil.

Jakob Nielsen, gurú de la usabilidad web, hace un análisis en su libro “Usabilidad en dispositivos móviles” de los dispositivos móviles que usamos a diario, cómo vemos el contenido y cómo interactuamos con éste. Lo primero para tener en cuenta: reducir la información secundaria. Debemos tener en cuenta que las pantallas móviles son mucho más reducidas, lo que hace que la comprensión de lectura tenga el doble de grado de complejidad. Nielsen plantea que la interfaz debe ser limpia y sencilla, que cubra sólo lo básico con textos cortos, directos y concisos.

Por este principio de usabilidad apreciamos como en la web de Tripadvisor tenemos bastante más opciones y texto que en la aplicación móvil.



Oshichi Teppanyaki

Calle Experimento no 7 | 03803, 03603 Benidorm, España

Tu experiencia es muy valiosa para otros viajeros. ¡Muchas gracias!

Tu puntuación general de este restaurante

Haz clic para puntuar

Título de tu opinión

Resume tu visita o destaca algún detalle interesante

Tu opinión

Ideas para escribir una buena opinión

Cuenta tu experiencia a la gente: la comida, el ambiente, el servicio...

(100 caracteres mínimo)

¿Qué tipo de visita hiciste?

En pareja

En familia

Amigos

Negocios

En solitario

Estuviste aquí para...

Selecciona uno

¿Cuándo fuiste?

Selecciona uno

¿Nos puedes contar algo más? (opcional)

Sitio web de restaurante:

¿Este restaurante ofrece servicio de mesa?

Si

No

No estoy seguro/a

¿Una persona vegana podría comer bien en el restaurante?

Si

No

No estoy seguro/a

¿Una persona vegetariana podría comer bien en el restaurante?

Si

No

No estoy seguro/a

¿Ofrece este restaurante sillas altas para niños?

Si

No

No estoy seguro/a

¿Se podría considerar este sitio un mercado de alimentos especializado?

Si

No

No estoy seguro/a

¿Es principalmente una panadería?

Si

No

No estoy seguro/a

Haz clic para seleccionar una puntuación

Servicio


Haz clic para puntuar

Comida

Haz clic para puntuar

Relación calidad-precio

Haz clic para puntuar

¿Qué tal está de precio este restaurante? 

Comida barata

Gama media

Restaurantes elegantes

¿Qué platos recomendarías?

¿Tienes alguna foto que compartir? (opcional)

Añadir una foto

Envía tu opinión

☐ Certifico que esta opinión está basada en mi propia experiencia, que refleja mi opinión sincera sobre este restaurante, que no tengo ninguna relación, ya sea personal o empresarial, con el mismo y que además no me ha ofrecido incentivo o pago alguno por escribirla. Del mismo modo, entiendo que TripAdvisor tiene una política de tolerancia cero con las opiniones falsas. Obtén más información

Envía tu opinión

Vista previa de la opinión

JAZZTEL

12:59

85 %

Cancelar

Oku Sushi Teppanyaki

Enviar

Pulsa un círculo para puntuar

Mes de visita

jun de 2018

Tipo de visita

Título

Opinión

Enviar

De Tripadvisor podemos destacar la cantidad de información que ofrece tanto de restaurantes como de opiniones de usuarios, ya que dispone de una comunidad bastante grande y activa. La cantidad de filtros que tiene a la hora de buscar un restaurante puede ser abrumadora, pero a la vez es muy útil ya que nos permite acotar en base a diversos factores.

### 2.1.3. Yelp

Tiene un buscador con diversas opciones que no solo nos permite buscar restaurantes, sino también lugares de copas, locales que sirvan a domicilio e incluso servicios locales como cerrajeros, fontaneros, etc.

También podemos encontrar en su página principal una sección donde podemos ver las últimas reseñas de los usuarios.

En la ficha de los locales encontramos su información, reseñas y fotos subidas por los usuarios.

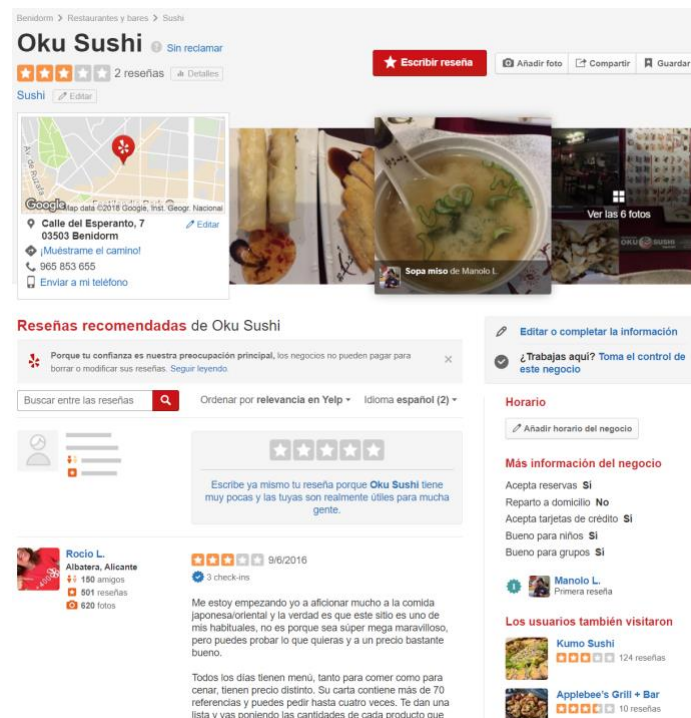


Figura 7. Yelp - Ficha de un restaurante  
(Fuente <https://www.yelp.es>)

Como característica, los usuarios pueden editar la información de los restaurantes, siendo esta verificada por el equipo de Yelp. Esto puede suponer un inconveniente, y es que buscando uno de mis locales favoritos en la aplicación, no lo encontraba. Después de indagar he visto que tienen el nombre mal, el local se llama Yamato y en la web aparece como Yamoto.

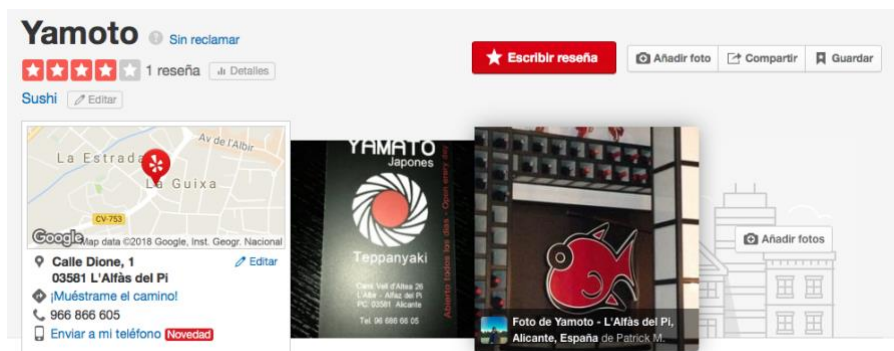


Figura 8. Yelp - Error en el nombre de un restaurante  
(Fuente <https://www.yelp.es>)

Al escribir una reseña nos redirige a una página en la que podemos puntuar mediante estrellas del 1 al 5 y describir nuestra experiencia.

## Oku Sushi

★ ★ ★ ★ ★

Selecciona tu puntuación

Tu reseña es útil porque ayuda a otras personas a conocer fantásticos pequeños comercios.

Por favor, no comentes este negocio si eres el propietario, un empleado o ex-empleado o si has recibido un regalo a cambio de escribir tu reseña.

Publicar reseña

[Lee nuestras directrices sobre reseñas](#)

Figura 9- Yelp - Escribir reseña  
(Fuente <https://www.yelp.es>)

A pesar de que Yelp nos ofrece tanto la información del restaurante, su posición en el mapa y opiniones de usuarios igual que Google y Tripadvisor, cuenta con una comunidad muchísimo más pequeña, al menos en la zona de Benidorm. Podemos ver en la Figura 2, Figura 3 y Figura 8 que para el mismo restaurante la cantidad de comentarios en Yelp es demasiado baja. Siendo los comentarios en Yelp un 1,13% del total de los comentarios en Tripadvisor.

Con respecto a la aplicación no se observan diferencias más allá de que se le ha dado una versión móvil.



#### 2.1.4. Facebook

En Facebook podemos encontrar las páginas de algunos restaurantes y escribir una opinión, así como ver las opiniones que otros usuarios han escrito.



*Figura 10. Facebook - Escribir opinión sobre un restaurante  
(Fuente <https://www.facebook.com>)*

Aunque lo que me inspiró de Facebook a hacer este proyecto fueron los grupos, y más concretamente aquellos en los que el tema principal es la restauración.

Un grupo es un sitio donde la gente deja sus opiniones, sus enlaces, comenta lo publicado por otros en torno a un tema común o simplemente lee el contenido que el resto de los usuarios ha publicado. Nos permite interactuar con gente (que no tiene por qué ser amiga nuestra en Facebook) pero que tiene los mismos intereses.

En este caso hablaré del grupo “Comer BBB (Bien, Bueno y Barato) - Benidorm y Provincia” del que soy miembro desde hace un par de años.

En el grupo la gente suele escribir reseñas de los sitios a los que ha ido acompañándola con fotos de los platos y del local. Además de compartir sus experiencias, hay usuarios que piden recomendaciones sobre restaurantes a los que ir, según la zona, tipo de comida o precio.

Los grupos suelen tener una plantilla predefinida para intentar que todas las crónicas cumplan con un formato específico. En concreto la plantilla que se suele usar en este grupo es la siguiente:

- NOMBRE LOCAL:
- DIRECCION:
- TELEFONO:
- TIPO DE COMIDA:
- PRECIO POR PERSONA:
- CRÓNICA





Figura 11. Facebook - Reseña de un usuario  
(Fuente <https://www.facebook.com/groups/bbbbenidorm/>)

Las reseñas aparecen ordenadas por fecha y la herramienta de la que disponemos para encontrar una reseña es un buscador mediante palabras clave. Por lo que no podemos agrupar por tipos de comida, precio, etc.

Facebook está disponible desde web y en aplicación, tanto para Android como para IOS y de forma gratuita, aunque el acceso a este tipo de grupos está restringido y el administrador tiene que aprobar el acceso.

## 2.2. Aplicaciones para realizar reservas

Estas aplicaciones tienen como principal servicio la reserva de mesas en los restaurantes, aunque también podemos ver opiniones de los usuarios.

### 2.2.1. ElTenedor

En la portada tenemos un buscador, un listado con los restaurantes de la zona que le hayamos indicado y un filtro para acotar el resultado.

El buscador podríamos dividirlo en tres partes, una donde indicamos si queremos restaurantes con promociones activas, restaurantes con buenas puntuaciones o restaurantes con un tipo de cocina concreto (1). Otra donde estableceremos nuestra localización e introduciremos la dirección o nombre del restaurante (2). Y la tercera, para seleccionar la fecha en la que acudiremos, la hora y el número de personas (3).

Tipo de restaurante 1

2 Dirección, nombre de restaur...

3 Fecha Hora Pers.

BUSCAR

**TIPO DE RESTAURANTE**

- TODO  
Todas las mesas
- INSIDER  
Las mejores mesas
- POP  
Las mejores promos

**TIPO DE COCINA**

Todo	Argentino	Asador	De mercado
De Fusión	Español	Francés	Indio
Internacional	Italiano	Japonés	Mediterráneo
Mexicano	Vasco		

Figura 12. ElTenedor – Buscador  
(Fuente <https://www.eltenedor.es>)

Al entrar en la ficha de un restaurante, lo primero que vemos es la puntuación, fotos de los usuarios, un calendario donde aparecen disponibles las fechas para reservar y si hay alguna promoción activa.

Guía de restaurantes / Restaurantes en Benidorm / Restaurante Venta L'Almartxa en Benidorm

**Venta L'Almartxa**

9,3/10  
211 opiniones

Por Avda Andalucía cruzando vía del tren 03502 Benidorm España

MAPA

MEDITERRÁNEO DE NEGOCIOS

Precio medio 25 €

**Reservar**  
Gratuito - inmediato - garantizado  
Hasta -30% en Cuenta Final

mié 20 jun

Junio 2018

lun	mar	mié	jue	vie	sáb	dom
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Premiamos tu confianza  
Con esta reserva ganarás 50 Yums

Figura 13. ElTenedor - Ficha de un restaurante  
(Fuente <https://www.eltenedor.es>)

También podemos ver en la ficha opiniones de los usuarios de ElTenedor y además opiniones procedentes de Tripadvisor ya que desde 2014 ElTenedor pertenece a Tripadvisor. Quiero destacar que para hacer una reseña en ElTenedor es necesario haber reservado a través de su web o aplicación.



Figura 14. ElTenedor – Opiniones  
(Fuente <https://www.eltenedor.es>)

Además de la información detallada del restaurante, dispone de un apartado donde podemos ver algunos platos de la carta con sus precios.

Carta del restaurante Venta L'Almartxa	
<b>Entrantes</b>	
Ensalada Almartxa Mix de lechugas, queso feta, tomates cherry, fruta de temporada, frutos secos, vinagreta mostaza y miel	3,5 €
Ensalada César Lechuga, pollo, queso parmesano, picatostes y salsa César	4 €
Ensalada Chef Natalia Mix de lechugas, bacon frito, queso de cabra tostado, coulis de frutos rojos, tomate	3,5 €
VER MÁS ▼	
<b>Plato Principal</b>	
Pollo frito con tomate, o con ajos, también posible a la brasa precio medio pollo con patatas fritas	8,5 €
Conejo frito con tomate o ajos, también posible a la brasa precio medio conejo con patatas fritas	11 €
Magret de Pato a la brasa con salsa chutney de mango,	17,5 €
Brocheta de Pollo rellena de queso crema envuelta con bacon y con piña	10,5 €
Steak Tartare	19,95 €

Figura 15. ElTenedor - Carta del restaurante  
(Fuente <https://www.eltenedor.es>)

Con respecto a la aplicación podemos encontrar la misma información que en web, pero con un diseño adaptado a móvil, mas simplificado y con la información secundaria en forma de submenús.

He de destacar que en ElTenedor solo aparecerán los restaurantes afiliados al sistema de reservas. En el caso de Benidorm son 91 restaurantes de los 783 que aparecían en Tripadvisor.

## 2.3. Aplicaciones para pedir comida a domicilio

### 2.3.1. Just Eat

Lo primero que tenemos que introducir al entrar en Just Eat es una dirección para que nos pueda ofrecer los restaurantes que reparten en nuestra zona.

Los resultados son mostrados en forma de lista con su puntuación, la dirección y el horario de apertura si estuviera cerrado.

Podemos filtrar estos resultados por tipo de comida y también por filtros especiales como ofertas activas, sin gastos de envío, con 5 estrellas, etc.

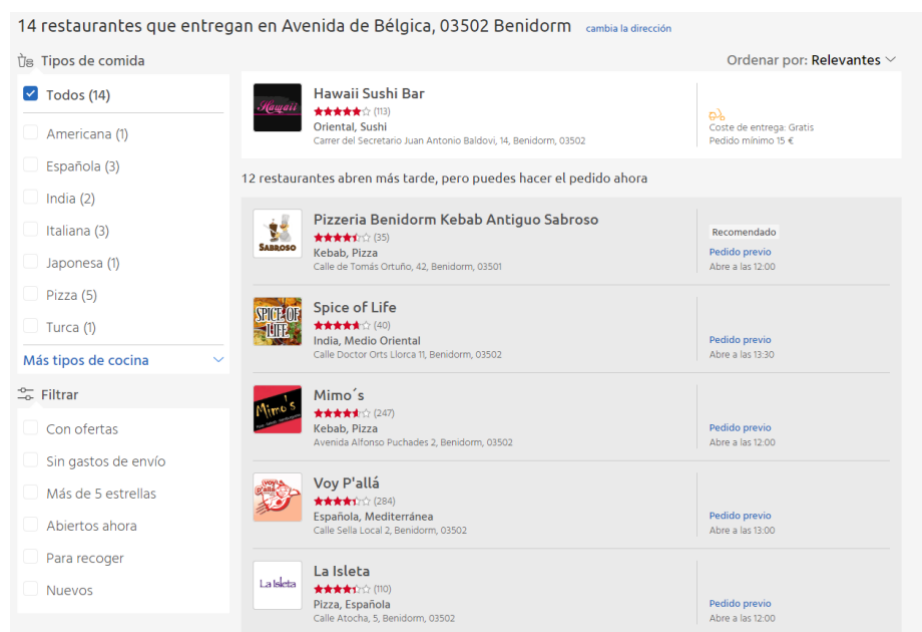


Figura 16. Just Eat - Resultado de la búsqueda  
(Fuente <https://www.just-eat.es>)

Al entrar en la ficha de un restaurante, lo primero que vemos son los platos del restaurante y un menú con las categorías en las que están divididos. Tenemos también una pestaña con las opiniones de los usuarios y otra con información sobre el restaurante.

En la ficha del restaurante también hay un carrito donde podemos ver los platos que hemos añadido y el total a pagar. Una vez terminado podemos elegir si queremos que nos lo envíen a casa o recogerlo en el local si dispone de esa opción.

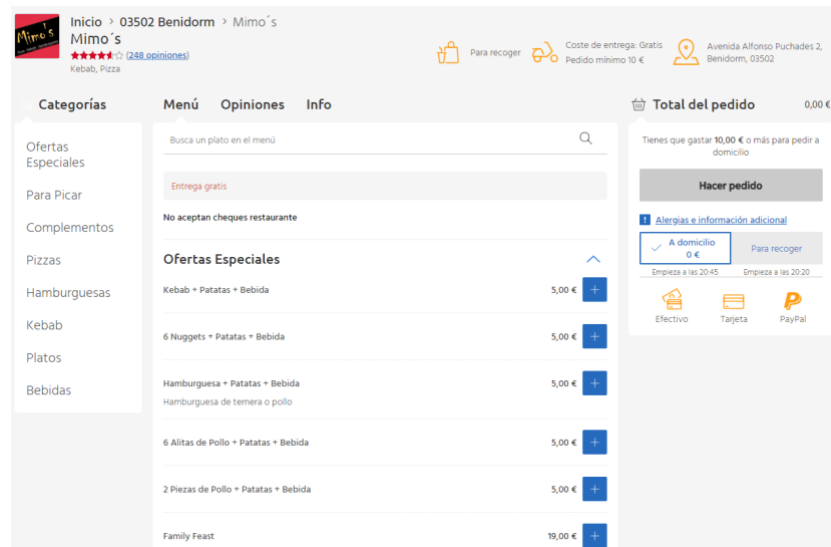


Figura 17. Just Eat - Ficha de un restaurante  
(Fuente <https://www.just-eat.es>)

Como pasaba con ElTenedor, en Just Eat tampoco podemos escribir una opinión si no hemos hecho un pedido a través de la web o aplicación.

La interfaz es simple e intuitiva tanto en la versión web como en la aplicación y en pocos pasos se puede hacer un pedido.

Gracias al estudio de las distintas aplicaciones existentes, hemos podido ver las diferentes soluciones que ofrecen para el problema que planteábamos en la introducción.

En todas las aplicaciones disponemos de un buscador con distintas opciones que ayudarán al usuario a encontrar el restaurante ideal. En la ficha de los restaurantes podemos encontrar la información detallada del restaurante, las opiniones de los usuarios y la opción para poder escribir nuestra propia reseña. Otra de las opciones que tienen en común es que nos dan la posibilidad de poder guardarnos en favoritos los restaurantes para tener una lista personalizada, aunque en ninguna podíamos clasificarlos en varias listas como restaurantes favoritos y por ejemplo otra lista con restaurantes que te gustaría probar.

Una de las principales carencias que destacaría que tienen las aplicaciones analizadas y que solo resuelve Facebook mediante los grupos mencionados, es la posibilidad de visualizar los

diferentes comentarios de los usuarios sin la necesidad de tener que entrar uno a uno a cada restaurante. En este proyecto nos centraremos en esta característica sin dejar de lado las anteriores mencionadas.

### 3. Objetivos

El objetivo principal de este proyecto es el de crear una plataforma para dar soporte a una red social temática en la cual los usuarios puedan compartir sus experiencias en los restaurantes que han visitado, descubrir nuevos locales a los que ir y consultar información de los restaurantes cercanos.

- Sub-objetivo 1: Elaborar el estado del arte de forma completa analizando las distintas páginas webs y aplicaciones móvil existentes que sean similares a lo que se va a crear.
- Sub-objetivo 2: Realizar un análisis funcional basado en las necesidades para esta aplicación. Identificar los requisitos funcionales y no funcionales.
- Sub-objetivo 3: Diseño del sistema tanto a nivel de arquitectura de la aplicación y los modelos de datos como el diseño de las interfaces y la interacción.
- Sub-objetivo 4: Desarrollar el sistema propuesto en este proyecto atendiendo las siguientes directrices:
  - Sub-objetivo 4.1: Ofrecer contenido preciso y completo. Para ello se buscará que la información de los restaurantes se obtenga de fuentes externas. De esta forma se añadirá exactitud, corrección y homogeneidad a la aplicación.
  - Sub-objetivo 4.2: Hacer que el sistema sea multiplataforma y accesible desde cualquier parte.
  - Sub-objetivo 4.3: Crear un backoffice para poder moderar las reseñas de los usuarios.
  - Sub-objetivo 4.4: El sistema permitirá distintos roles de usuario. Administrador, usuario identificado y usuario anónimo.
- Sub-objetivo general: Aprender a tomar decisiones, adquirir experiencia, aprender y perfeccionar nuevas tecnologías.

## 4. Metodología

A la hora de elegir una metodología tenemos que saber que no hay una metodología mejor que otra, sino metodologías que tienen diferentes usos y enfoques.

Voy a analizar las metodologías Ágiles y el modelo en Cascada, para ver cuál es la que mejor se adaptará al proyecto.

La idea básica de una metodología Ágil es que es interactiva y cíclica, lo que significa que la implementación del software ocurre de forma incremental. Dentro de las metodologías Ágiles tenemos varias como Scrum, Kanban, Programación Extrema, etc. Me voy a centrar en Scrum ya que es la metodología con la que estoy más familiarizado por haberla dado en la carrera.

En esta metodología, en vez de crear todas las tareas y los horarios por adelantado, todo el tiempo es encajonado en fases llamadas sprints. Cada sprint tiene una duración definida, normalmente de 2 a 4 semanas, con una lista de ejecución con entregables, y con cada sprint planificado por adelantado. Los entregables son priorizados por su valor empresarial el cual es determinado por el cliente. Si todo el trabajo planificado para un sprint no puede ser completado, se le vuelve a dar prioridad al trabajo y la información se utiliza para la planificación de los sprints futuros.



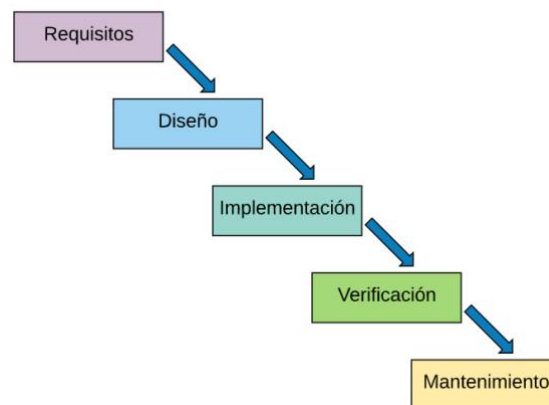
*Figura 18. Proceso de desarrollo Scrum*

(Fuente: <http://www.penadelarosa.com/concepto-de-moda/scrum-en-un-post/> )

A medida que el trabajo es completado durante cada sprint, se revisa y evalúa continuamente por el cliente. Todo el trabajo que ha sido completado debe ser definido como entregable, lo que significa que debe funcionar como se debe y debe haber sido testeado. El ciclo continúa hasta que el producto haya sido completado y esté listo para ser desplegado o sea considerado como aceptable por la empresa.



El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra.



*Figura 19. Etapas del modelo en cascada*

(Fuente <https://openclassrooms.com/courses/gestiona-tu-proyecto-de-desarrollo/en-que-consiste-el-modelo-en-cascada>)

En este modelo, la etapa siguiente empieza cuando termina la anterior. Para ello primero se determinan las características del software a desarrollar y se especifica todo lo que debe hacer el sistema sin entrar en detalles técnicos, que sería la fase de requisitos. En la fase de diseño se describe la estructura interna del software, y las relaciones entre las entidades que lo componen. La siguiente fase es la implementación y es en la que se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior. Una vez terminada la implementación pasamos a la fase de verificación, donde se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos. Por último, una vez se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionan correctamente, se inicia la fase de instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcione correctamente en el entorno en que se va a utilizar.

La metodología que voy a seguir será el modelo en Cascada porque me permite atender toda la incertidumbre que hay en cada fase y no pasar a la siguiente hasta haber completado la anterior, mientras que en scrum el valor está en un equipo que se apoya mutuamente, al estar solo puedo cometer errores al pasar de una iteración a otra, lo que produciría retardos que paralizarían el proyecto, ya que es un proyecto nuevo para mí y hay mucha incertidumbre.

En los siguientes puntos serán detallados los bloques que lo conforman.

## 5. Análisis y especificación

### 5.1. Descripción general

En esta sección se definirán las funcionalidades y requisitos del proyecto. Para ello se ha decidido utilizar el estándar IEEE 830. Este estándar nos permitirá seguir un modelo para definir el proyecto y asegurarnos de que se abordarán todas las áreas del proyecto, desde los usuarios a las funcionalidades, analizando todos los aspectos que involucran al proyecto.

#### 5.1.1. Perspectiva del producto

El sistema utilizará APIs de terceros para la obtención de información sobre los restaurantes, lo que nos permitirá que la información sea veraz y no requiera de mantenimiento en ese aspecto.

#### 5.1.2. Funciones del producto

El sistema deberá mostrar las reseñas de los usuarios, así como también proporcionar un listado de restaurantes, teniendo siempre en cuenta la posición del usuario para ofrecer los resultados según la cercanía. Para ellos se separarán los listados en restaurantes y opiniones.

También se podrán buscar restaurantes, palabras claves en valoraciones y tipos de cocina con la posibilidad de filtrar los resultados para una mejor búsqueda. Los filtros deberán ser por valoración, relevancia, tipo de cocina y distancia.

Con respecto a los restaurantes, se podrá consultar información detallada de estos haciendo uso de una API externa, opiniones de los usuarios y valoración media.

Los usuarios podrán navegar por el sistema sin necesidad de identificarse. Si el usuario se ha identificado podrá crear reseñas sobre restaurantes para compartirlas con los demás usuarios, además de crear y gestionar una lista con sus restaurantes favoritos.

#### 5.1.3. Características de los usuarios

Los principales usuarios de la aplicación serán usuarios que estén interesados en encontrar nuevos restaurantes o compartir sus experiencias de los restaurantes a los que ha ido con el resto de usuario y su nivel educativo o de conocimientos no tiene que ser necesariamente elevado. Existirán dos tipos de usuarios, el administrador que se encargará de verificar si el contenido es apto para estar en la aplicación; y los usuarios de la aplicación.

#### 5.1.4. Restricciones

- **Costes económicos:** La principal restricción de este proyecto serán los costes económicos. Los costes de los servidores o acceso a servicios de terceros serán de tipo variable, por lo que no supone un problema al principio, pero si algo a tener en cuenta en el producto final en el caso de que este empiece a escalar. El presupuesto inicial será de 50€ para la cuenta de desarrollador de Android y el servidor para alojar el sistema.
- **Seguridad:** Por un lado, se implementará una identificación de usuario para realizar ciertas acciones y, por otro lado, una validación por token para realizarlas.

#### 5.1.5. Suposiciones y dependencias

Una parte de la funcionalidad de la aplicación dependerá en gran medida de una API de terceros, por lo que el correcto funcionamiento de esta irá ligada a la API.

### 5.2. Especificación y requisitos

En este apartado dividiremos los requisitos en funcionales, no funcionales y de diseño, según describe el estándar IEEE 830.

#### 5.2.1. Requisitos funcionales

Cada requisito tendrá un identificador único; el actor involucrado, que en nuestro caso podrá ser el usuario anónimo, usuario identificado o el administrador; el nombre, que será un nombre descriptivo; la descripción del requisito; los requisitos lógicos y la prioridad.

Identificador	RF-01
Actor involucrado	Usuario identificado y usuario anónimo
Nombre	Mostrar reseñas
Descripción	El sistema mostrará las reseñas de los locales más cercanos a la posición del usuario.
Requisitos lógicos	El usuario deberá estar geolocalizado o deberá haber indicado una localización.
Prioridad	Alta

Identificador	RF-02
Actor involucrado	Usuario identificado y usuario anónimo
Nombre	Mostrar restaurantes
Descripción	El sistema mostrará los restaurantes más cercanos a la posición del usuario.
Requisitos lógicos	El usuario deberá estar geolocalizado o deberá haber indicado una localización.
Prioridad	Alta

Identificador	RF-03
Actor involucrado	Usuario identificado y usuario anónimo
Nombre	Buscador
Descripción	A partir de una cadena de texto y utilizando un buscador, el usuario podrá encontrar restaurantes y reseñas que contengan dicha cadena.
Requisitos lógicos	Debido a la naturaleza de los posibles resultados, el sistema dividirá los datos en restaurantes y reseñas.
Prioridad	Alta

Identificador	RF-04
Actor involucrado	Usuario identificado y usuario anónimo
Nombre	Localización
Descripción	El usuario deberá estar geolocalizado o definir manualmente una localización.
Requisitos lógicos	El sistema preguntará al usuario si da su consentimiento para estar geolocalizado mientras utiliza la aplicación.
Prioridad	Alta

Identificador	RF-05
Actor involucrado	Usuario identificado y usuario anónimo
Nombre	Ver información de un restaurante
Descripción	El sistema mostrará la información sobre los restaurantes y las reseñas asociadas.
Requisitos lógicos	Habrà una serie de datos mínimos y comunes a mostrar en cada restaurante como es la información básica del restaurante y su valoración.
Prioridad	Alta

Identificador	RF-06
Actor involucrado	Usuario identificado
Nombre	Valorar restaurante
Descripción	Se permitirá a los usuarios que hayan iniciado sesión valorar los restaurantes.
Requisitos lógicos	El sistema permitirá introducir una valoración numérica, adjuntar imágenes y escribir un texto. sin ningún tipo de restricción para valorar excepto la de introducir la valoración numérica.
Prioridad	Alta

Identificador	RF-07
Actor involucrado	Usuario anónimo
Nombre	Identificación
Descripción	Cualquier usuario podrá identificarse en la aplicación.
Requisitos lógicos	La identificación se hará a través de Google, lo que nos devolverá un token único que lo identificará. Este token será enviado en la cabecera de las peticiones HTTP que requieran autenticación.
Prioridad	Alta

Identificador	RF-08
Actor involucrado	Usuario identificado
Nombre	Cerrar sesión
Descripción	Un usuario que haya iniciado sesión debe ser capaz de cerrar sesión en cualquier momento.
Requisitos lógicos	Al cerrar sesión el sistema borrará el token generado.
Prioridad	Alta

Identificador	RF-09
Actor involucrado	Usuario identificado
Nombre	Gestionar restaurantes
Descripción	El sistema permitirá marcar o desmarcar un restaurante como favorito y/o como restaurante para ver más tarde y ver un listado de estos.
Requisitos lógicos	
Prioridad	Alta

Identificador	RF-10
Actor involucrado	Usuario identificado y usuario anónimo
Nombre	Ver perfil de usuario
Descripción	El sistema permitirá consultar la información de los usuarios.
Requisitos lógicos	Se podrán ver las reseñas que el usuario ha hecho y sus restaurantes favoritos.
Prioridad	Alta

Identificador	RF-11
Actor involucrado	Usuario administrador
Nombre	Administrar contenido
Descripción	El sistema permitirá al usuario con rol de administrador decidir si las opiniones son aptas o no para estar en la aplicación.
Requisitos lógicos	Los usuarios con rol de administrador tendrán otra vista para poder moderar las reseñas y aceptarlas o rechazarlas.
Prioridad	Media

### 5.2.2. Requisitos no funcionales

#### **RNF-01: Ubicuidad**

El sistema deberá implementarse pensando que el dispositivo del usuario será el móvil, para permitir el fácil acceso desde cualquier parte.

#### **RNF-02: Escalabilidad**

La aplicación debe ser escalable, teniendo la capacidad de crecer sin desmejorar la calidad del servicio. El sistema debe ser capaz de crear nuevas instancias para balancear la carga.

#### **RNF-03: Rendimiento**

El sistema de procesar las operaciones de un usuario individual dentro de unos tiempos de respuesta inferiores a 3 segundos.

#### **RNF-04: Disponibilidad**

La aplicación deberá estar disponible la mayor cantidad de tiempo posible, para que los usuarios puedan acceder a ella. El servicio de hosting que se use deberá ofrecer una disponibilidad de casi un 100%.

#### **RNF-05: Robustez**

La aplicación debe continuar en operación a pesar de la entrada de datos inválidos o fallos en los diferentes componentes que lo conforman.

## RNF-06: Seguridad

Todas las peticiones al sistema harán uso del protocolo HTTPS. Los usuarios se identificarán en la aplicación usando un token generado por la API siguiendo el protocolo OAuth 2.0.

### 5.2.3. Requisitos deseables

Identificador	RF-14
Actor involucrado	Usuario identificado y usuario anónimo
Nombre	Mostrar restaurantes en un mapa
Descripción	El sistema debe ser capaz de mostrar los restaurantes más cercanos a la posición del usuario gráficamente en un mapa.
Requisitos lógicos	El usuario deberá estar geolocalizado o deberá haber indicado una localización.
Prioridad	Baja

Identificador	RF-15
Actor involucrado	Usuario identificado
Nombre	Gamificación del sistema
Descripción	Incluir un sistema de logros para incentivar el escribir reseñas.
Requisitos lógicos	Recompensar al usuario por escribir reseñas y que sea visible por los demás usuarios.
Prioridad	Baja

Identificador	RF-16
Actor involucrado	Usuario identificado
Nombre	Sistema de reseña útil
Descripción	El sistema permitirá indicar si una reseña ha sido útil o no.
Requisitos lógicos	
Prioridad	Baja



Identificador	RF-17
Actor involucrado	Usuario identificado
Nombre	Seguir usuarios
Descripción	El sistema permitirá seguir a otros usuarios, ver un listado de usuarios que seguimos y que nos siguen.
Requisitos lógicos	
Prioridad	Media

Identificador	RF-18
Actor involucrado	Usuario anónimo
Nombre	Identificación con Facebook
Descripción	Cualquier usuario podrá identificarse en la aplicación usando Facebook.
Requisitos lógicos	La identificación se hará a través de Facebook, lo que nos devolverá un token único que lo identificará. Este token será enviado en la cabecera de las peticiones HTTP que requieran autenticación.
Prioridad	Alta

## 6. Diseño

El diseño de la aplicación incluye tanto el diseño de la arquitectura de la aplicación como el diseño de las interfaces y la interacción.

### 6.1. Diseño de la persistencia

La persistencia requiere que los datos sean almacenados en un medio secundario, no volátil, para su posterior reconstrucción y utilización, por lo que su tiempo de vida es independiente del proceso que los creó.

En nuestro proyecto usaremos una base de datos no relacional o NoSQL. El término NoSQL aparece con la llegada de la web 2.0, ya que hasta ese momento sólo subían contenido a la red aquellas empresas que tenían un portal, pero con la llegada de aplicaciones como Facebook, Twitter o Youtube, cualquier usuario podía subir contenido, provocando así un crecimiento exponencial de los datos.

Empresas como Google, Amazon, Twitter y Facebook vieron que con el crecimiento de la web en tiempo real existía una necesidad de proporcionar información procesada a partir de grandes volúmenes de datos que tenían unas estructuras horizontales más o menos similares. Estas compañías se dieron cuenta de que el rendimiento y sus propiedades de tiempo real eran más importantes que la coherencia, en la que las bases de datos relacionales tradicionales dedicaban una gran cantidad de tiempo de proceso.

Al tratarse mi proyecto de una red social, una base de datos NoSQL se adaptará perfectamente y además me dará la libertad para experimentar y equivocarme, ya que cuando esté prototipando, los cambios serán más rápidos y tendrán menos consecuencias.

Otras de las ventajas que ofrecen las bases de datos NoSQL es que pueden ejecutarse en máquinas con pocos recursos ya que no requieren apenas de computación, lo que abarata costes si tenemos en cuenta que una de nuestras restricciones era el presupuesto.

Hemos optado por usar una base de datos no relacional y en concreto MongoDB por todas las ventajas que ofrece y además porque durante el ABP vimos un poco esta base de datos y como uno de los objetivos de este proyecto es aprender y perfeccionar nuevas tecnologías, creo que es una buena oportunidad.

Como este proyecto se alimentará de una API externa para obtener la información de los restaurantes, los documentos que almacenaremos en la base de datos serán de tres tipos: usuario, comentario e imagen.

```
{
  "Comentario": {
    "_id": String::index,
    "contenido": String,
    "f_crea": Date,
    "f_modi": Date,
    "autor": Usuario,
    "placeid": String,
    "longitud": Number,
    "latitud": Number
  }
}

{
  "Usuario": {
    "_id": String::index,
    "nombre": String,
    "apellidos": String,
    "email": String,
    "rol": String,
    "imagen": Imagen
  }
}

{
  "Imagen": {
    "_id": String::index,
    "comentario": Comentario
  }
}
```

*Figura 20. Modelo de datos*

En la Figura 21 podemos apreciar que no se han definido relaciones entre los documentos ya que la base de datos usada no es relacional, pero como se puede ver, el documento “Comentario” tiene un atributo de tipo “autor”, cuyo tipo es del documento “Usuario”.

El esquema propuesto contiene los datos necesarios para cumplir con los requisitos de mayor prioridad, aunque puede ser susceptible de ser modificado más adelante.

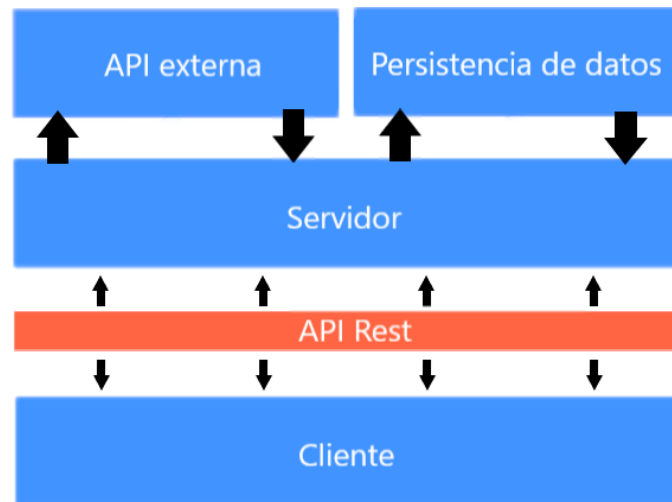
## 6.2. Diseño arquitectura conceptual

Como podemos ver en la Figura 22, la arquitectura de nuestra aplicación se dividirá en tres capas:

La capa de persistencia de datos que junto con la API externa nos proporcionarán los datos que se usarán en el resto de la aplicación.

El servidor que se comunicará con la capa de datos usando el modelo de abstracción y con el cliente usando una interfaz REST.

El cliente que interactuará con el servidor a través de la interfaz REST independientemente de la tecnología que usemos en el mismo.



*Figura 21. Esquema conceptual de la arquitectura*

### 6.3. Diseño API Rest

Nuestra API Rest estará compuesta por tres modelos de datos, uno por cada documento especificado como definíamos en la Figura 21. La flexibilidad que ofrece MongoDB nos permite crear los documentos e índices en nuestra base de datos definiendo los modelos.

Por cada modelo tendremos agrupados los controladores que interactuarán con los modelos y realizarán las acciones y operaciones sobre nuestra base de datos para finalmente generar una respuesta.

Así mismo, para cada controlador habrá una ruta vinculada que se encargará de decidir qué hacer con cada petición que reciba de nuestra aplicación. Estas rutas también estarán agrupadas por cada modelo de datos para una mejor estructuración del código de nuestro proyecto.

Por último, tendremos los servicios. Aquí se definirá toda funcionalidad que vaya a usarse frecuentemente o que no encaje dentro de los controladores, como el acceso a las APIs externas.

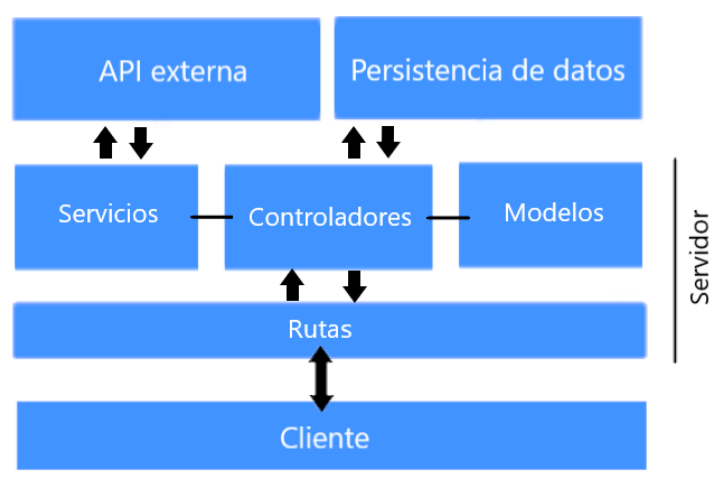


Figura 22. Arquitectura del servidor

Ahora que tenemos la estructura de nuestra API Rest, pasaré a definir las distintas rutas que harán de intermediario entre el Frontend y el Backend.

Si el recurso requiere de autorización, se añadirá [AUTH] a continuación de la ruta.

### 6.3.1. Rutas de comentario

Ruta	/comentario
Método	GET
Parámetros	<p>pagina: Página de resultados</p> <p>longitud: Coordenada de longitud del usuario</p> <p>latitud: Coordenada de latitud del usuario</p> <p>valoración: Valoración mínima que tienen que tener los resultados.</p>
Cuerpo del mensaje	
Descripción	<p>Obtiene los comentarios que se han hecho sobre restaurantes cercanos a las coordenadas pasadas por parámetro. Si el parámetro valoración no va vacío, las valoraciones deberán ser mayores o iguales al valor pasado. Si el parámetro página va vacío devolverá los 10 resultados y a medida que se vaya haciendo scroll en la aplicación se irán cargando comentarios de 10 en 10.</p>

Ruta	/comentario/restaurante/:id
Método	GET
Parámetros	id: Id del restaurante. pagina: Página de resultados
Cuerpo del mensaje	
Descripción	Obtiene los comentarios del restaurante en concreto. Si el parámetro página va vacío devolverá los 10 últimos resultados y a medida que se vaya haciendo scroll en la aplicación se irán cargando comentarios de 10 en 10.

Ruta	/comentario/restaurante/:id [AUTH]
Método	POST
Parámetros	id: Id del restaurante.
Cuerpo del mensaje	contenido: comentario del usuario f_crea: Fecha de creación del comentario f_modi: Fecha de la última modificación del comentario autor: Documento de tipo Usuario placeid: Id del restaurante devuelto por la API externa. longitud: Coordenada de longitud del restaurante latitud: Coordenada de latitud del restaurante
Descripción	Crea un comentario para el restaurante indicado.

Ruta	/comentario/usuario/:id
Método	GET
Parámetros	id: Id del usuario. pagina: Página de resultados
Cuerpo del mensaje	
Descripción	Obtiene los comentarios de un usuario en concreto. Si el parámetro página va vacío devolverá los 10 últimos resultados y a medida que se vaya haciendo scroll en la aplicación se irán cargando comentarios de 10 en 10.

Ruta	/comentario/:id [AUTH]
Método	PUT
Parámetros	id: Id del comentario.
Cuerpo del mensaje	contenido: comentario del usuario f_modi: Fecha de la última modificación del comentario
Descripción	Edita el comentario.

Ruta	/comentario/:id [AUTH]
Método	DELETE
Parámetros	id: Id del comentario.
Cuerpo del mensaje	
Descripción	Elimina el comentario.

### 6.3.2. Rutas de imágenes

Ruta	/imagen/comentario/:id
Método	GET
Parámetros	id: Id del comentario.
Cuerpo del mensaje	
Descripción	Obtiene las imágenes asociadas al comentario.

Ruta	/imagen/comentario/:id [AUTH]
Método	POST
Parámetros	id: Id del comentario.
Cuerpo del mensaje	comentario: Documento de tipo comentario
Descripción	Añade la imagen al comentario.

Ruta	/imagen/:id
Método	GET
Parámetros	id: Id de la imagen.
Cuerpo del mensaje	

Descripción	Devuelve la imagen asociada al ID pasado por parámetro.
-------------	---

Ruta	/imagen/:id [AUTH]
Método	DELETE
Parámetros	id: Id de la imagen.
Cuerpo del mensaje	
Descripción	Elimina la imagen.

### 6.3.3. Rutas de usuario

Ruta	/usuario/:id
Método	GET
Parámetros	id: Id del usuario.
Cuerpo del mensaje	
Descripción	Devuelve toda la información del usuario.

Ruta	/usuario/login
Método	GET
Parámetros	N / A
Cuerpo del mensaje	
Descripción	Identifica al usuario a través de la API de Google y recoge la información necesaria almacenándola en un token.

Ruta	/usuario/registro
Método	POST
Parámetros	token: Token obtenido a través de la API usada para identificar al usuario.
Cuerpo del mensaje	
Descripción	Una vez el usuario se haya identificado a través de la API, esta nos devolverá un token que usaremos para obtener los datos del usuario y registrarlo si es la primera vez.



Ruta	<b>/usuario/:id</b>
Método	PUT
Parámetros	id: Id del usuario
Cuerpo del mensaje	
Descripción	Modifica el alias que se mostrará del usuario en la aplicación.

#### 6.3.4. Identificación mediante Google OAuth API

Para identificar al usuario en nuestra aplicación usaremos la API de Google que utiliza el protocolo OAuth 2.0 para la autenticación y autorización. Para ello registraremos nuestra aplicación desde la consola de desarrolladores de Google creando el proyecto EatCom y obteniendo así el token de cliente.

En la Figura 24 podemos observar el flujo que seguirá la identificación del usuario. Para entender mejor el flujo explicaré que se envía y que nos devuelve Google.

Authorization code: Es una cadena de texto que nos devolverá Google, pero que no nos permite acceder a sus servicios, sino solicitar el token (access o refresh)

Token: Se trata de una cadena larga con un hash que nos devolverá Google. A su vez se dividen en:

Access Token: Token que servirá para un acceso normal de nuestra aplicación a los servicios. Generalmente caducará a la hora de ser expedido.

Refresh Token: Token que permitirá generar access tokens posteriormente de forma que nuestra aplicación no nos pedirá más permiso.

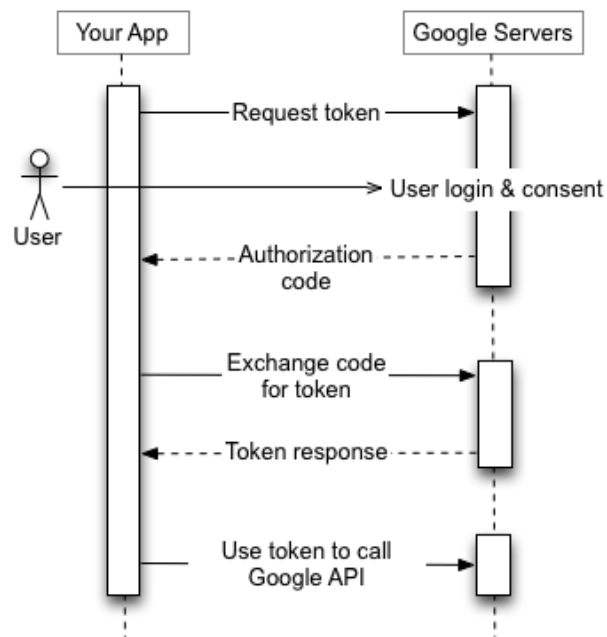


Figura 23. Flujo para identificar un usuario

(Fuente <https://www.ladrapalera.com/es/drupal/development/javascript/como-usar-una-api-de-google-con-autenticacion-traves-de-oauth2>)

Se ha decidido usar este método para la identificación en lugar de crear un propio sistema de registro debido a que no es necesario almacenar contraseñas de usuario y así evitar posibles vulnerabilidades. Al no tener un formulario de registro, al usuario solo le toma 2 clicks registrarse en nuestra aplicación y de esta forma la información que obtenemos es fiable y homogénea.

#### 6.3.5. Google Places API Web Service

Para obtener los listados de restaurantes y su información haremos uso de Google Places API Web Service.

Google Places API Web Service es un servicio que devuelve información sobre sitios, definidos dentro de la API como establecimientos, ubicaciones geográficas o puntos de interés importantes, mediante el uso de solicitudes de HTTP. Podemos buscar sitios por proximidad o una cadena de texto, devolviendo una lista de sitios junto con su información en formato JSON.

Los parámetros obligatorios en cada petición son:

- key: la clave de API de nuestra aplicación que hemos registrado en la consola de desarrolladores de Google.
- location: la latitud y longitud alrededor de las cuales se debe obtener la información sobre los restaurantes.

## 6.4. Diseño arquitectura tecnológica Front/Backend

Cualquier API REST se puede construir con cualquier lenguaje, puesto que todos los lenguajes actualmente tienen soporte a JSON. La arquitectura de una aplicación basada en API Rest es indiferente del lenguaje y por tanto cualquier alternativa es suficiente para acometer una tarea de hacer un Backend que permita su reutilización para diferentes Frontends, por lo que al final nuestra elección tiene bastante flexibilidad.

### 6.4.1. Frontend

El frontend de nuestra aplicación estará compuesto por HTML, CSS y Javascript. La parte de Javascript será la encargada de hacer que nuestra aplicación sea interactiva y para ello nos ayudaremos de frameworks o librerías.

- **Angular:** Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.
- **React:** Es una librería Javascript de código abierto para crear interfaces de usuario en una sola página. Es mantenido por Facebook, Instagram y una comunidad de desarrolladores independientes y compañías. React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. React sólo maneja la interfaz de usuario en una aplicación; está construida únicamente para utilizar el patrón de diseño modelo–vista–controlador (MVC), y puede ser utilizada junto con otras bibliotecas de Javascript o más grandes MVC como AngularJS.
- **Vue:** Es un framework de JavaScript de código abierto para construir interfaces de usuario. Puede integrarse fácilmente en proyectos que usen otras librerías de JavaScript, porque está diseñado para ser adoptado de forma incremental. Además, Vue también permite usar funciones de framework para aplicaciones web, que permite desarrollar avanzadas aplicaciones de una sola página.

### 6.4.2. Backend

- **PHP:** Existen frameworks potentes como Symfony o Laravel capaces de acelerar mucho el desarrollo de un API. Además, hay muchos microframeworks que tienen como principal objetivo el desarrollo de APIs REST, desechando todo lo pesado y no tan útil de un framework convencional, como Lumen o Slim.
- **.NET:** En esta tecnología también existen diversas alternativas sencillas y rápidas para implementar un API. El propio Microsoft nos ofrece ASP.NET Web API, un complemento esencial para construir servicios HTTP especialmente pensados para REST.
- **Javascript:** Si hablamos de utilizar Javascript del lado del servidor, NodeJS es el candidato ideal. Dado las características de esta plataforma, se hace muy adecuado para el desarrollo RESTful dado que Node, al no ser bloqueante, puede atender a más usuarios de manera concurrente. En NodeJS hay frameworks potentes para el desarrollo de APIs y aplicaciones en general, como Express o mongoose.

En relación con el RNF-01 se ha optado por desarrollar la aplicación con Cordova, que nos permite utilizar las tecnologías estándar web como HTML5, CSS3 y Javascript para desarrollo multiplataforma. Para ello hemos optado por usar Angular para el frontend y Node para el back, debido a ya tengo una base en esas tecnologías y quiero ampliar conocimientos y optar por otras podría suponer un riesgo a la hora de terminar el proyecto a tiempo.

Nuestro stack quedaría así:

- **MongoDB** para la gestión de bases de datos.
- **Express**, complemento para Node que se encargará de la gestión de peticiones y respuestas, cabeceras, rutas, vistas, etc.
- **Angular** para el desarrollo de la capa del cliente.
- **Node** para el desarrollo de la capa del servidor.

## 6.5. Diseño de la Experiencia de Usuario

La experiencia de usuario o UX se refiere a lo que la persona percibe cuando usa nuestra aplicación, es decir, cómo reacciona ante ella. Uno de los principales objetivos es conocer el público objetivo de nuestra aplicación, para poder crear una experiencia acertada. Nuestro público principal serán usuarios que quieran encontrar nuevos restaurantes y para ello busquen una aplicación que les ayude. La mayoría de estos usuarios estarán familiarizados con Facebook,

Twitter o Instagram, por lo que ofrecerles una interfaz con características similares a estas redes sociales hará que nuestra aplicación sea para el usuario mucho más intuitiva.

Un ejemplo sería el menú de una aplicación. En estos últimos años hemos visto como han ido desapareciendo los menús hamburguesa, que en un principio fueron revolucionarios por la cantidad de espacio que ahorraban en pantalla y redefinieron la arquitectura del contenido en las aplicaciones. Pero la larga, esta ventaja se convirtió en su punto débil porque a los usuarios les resultaba difícil encontrar ciertas opciones, haciendo que la navegación por la aplicación fuera una experiencia tediosa. Facebook o Youtube rectificaron y ahora cuentan con menús sencillos donde las opciones principales se encuentran siempre a la vista.



*Figura 24. Ejemplo de cambio de menú*  
(Fuente <http://jesusmartin.eu/muerte-menu-hamburguesa>)

Con este tipo de menú le damos al usuario la posibilidad de navegar a cualquier sección de la aplicación independientemente de la sección en la que se encuentre, ofreciendo una velocidad de navegación mucho mayor.

Si hablamos de ofrecer una buena experiencia en una aplicación donde el objetivo es encontrar un restaurante, para ello necesitaremos un buscador. De forma inconsciente, los usuarios se han acostumbrado a navegar mediante búsquedas, tal vez por los 20 años que lleva Google en marcha. Además, en muchas ocasiones llevamos prisa y navegar por las opciones de filtrado puede ser tedioso, por lo que un buscador accesible desde cualquier sección se convertirá en nuestra mejor herramienta.

Una buena experiencia de usuario se basa también en los tiempos de carga de la aplicación, en nuestro caso, aplicaremos lo que se conoce como “Lazy Load” o carga diferida, que consiste en retrasar la carga o inicialización de un objeto hasta el momento de su utilización. Los comentarios o restaurantes se irán cargando en bloques conforme el usuario vaya haciendo scroll en la aplicación.

## 6.6. Diseño Interfaces

El diseño de interfaces o del frontend, ha de cumplir dos objetivos: ser fácil de usar y fácil de entender. Con estas dos premisas junto a los requisitos funcionales y el estudio anterior sobre el diseño de la experiencia de usuario se han diseñado los siguientes wireframe.

En la Figura 25 tenemos la sección principal de nuestra aplicación, que satisface los requisitos funcionales RF-01 y RF-02 y que podríamos dividir en cabecera, contenido y menú.

En la cabecera de la aplicación tenemos un campo para el buscador, un botón para determinar cómo filtramos la información que estamos viendo y un botón para establecer nuestra localización, de forma manual o con la geolocalización del dispositivo.

En el contenido tenemos la opción de ver las reseñas de los restaurantes cercanos a nuestra posición o un listado de los restaurantes cercanos.

El menú se ha simplificado lo máximo posible para que el usuario tenga las funciones principales al alcance y que pueda en todo momento saber en qué sección se encuentra y desplazarse por la aplicación fácilmente.



Figura 25. Inicio mostrando reseñas y restaurantes

Al usar el buscador que satisface el requisito RF-03, nos aparecerá un listado con las últimas búsquedas efectuadas devolviéndonos el sistema, por un lado, las reseñas de los usuarios que contengan la palabra clave buscada y, por otro lado, un listado con los restaurantes que la API de Google Place nos devuelva.

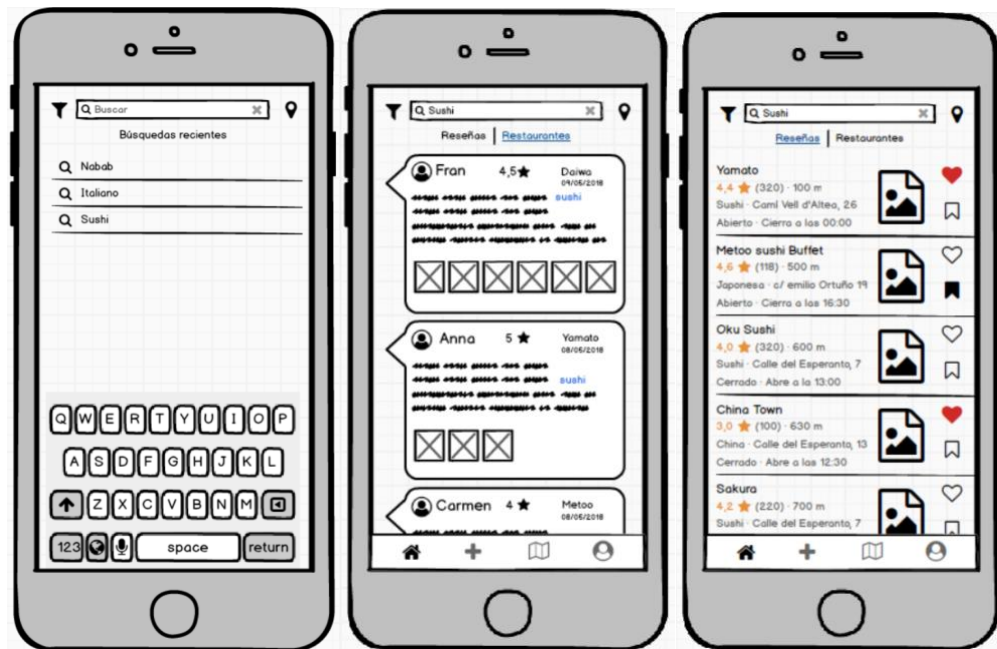


Figura 26. Buscador con resultados

Al pulsar en el botón de localización comentado anteriormente se nos abrirá un modal con un buscador en el cual podremos establecer nuestra posición de forma manual o seleccionando la posición que nos proporciona el dispositivo, satisfaciendo así el requisito RF-04.

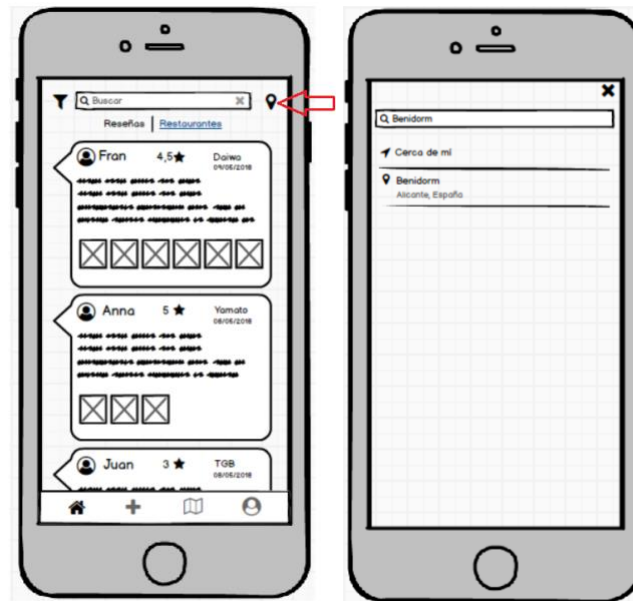


Figura 27. Localización

Volviendo al listado de restaurantes, al hacer click sobre uno de los restaurantes o sobre el nombre del restaurante en la reseña de un usuario, nos llevará a su ficha donde podremos ver la información del restaurante y todas sus reseñas asociadas, representando así el requisito RF-05.

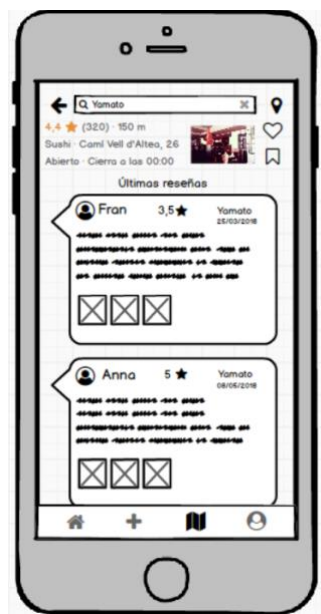


Figura 28. Ficha de un restaurante



Los usuarios pueden navegar libremente por la aplicación, pero para poder valorar restaurantes y gestionar sus listas deberán identificarse. Para identificarnos en la aplicación podremos acceder desde el icono de valorar restaurante o desde el icono de perfil si aún no nos hemos identificado. El usuario se identificará usando el inicio de sesión con Google, aunque en este diseño se ha incluido también el botón para iniciar sesión con Facebook, que es uno de nuestros requisitos deseables. En este esquema queda plasmado el requisito RF-07.

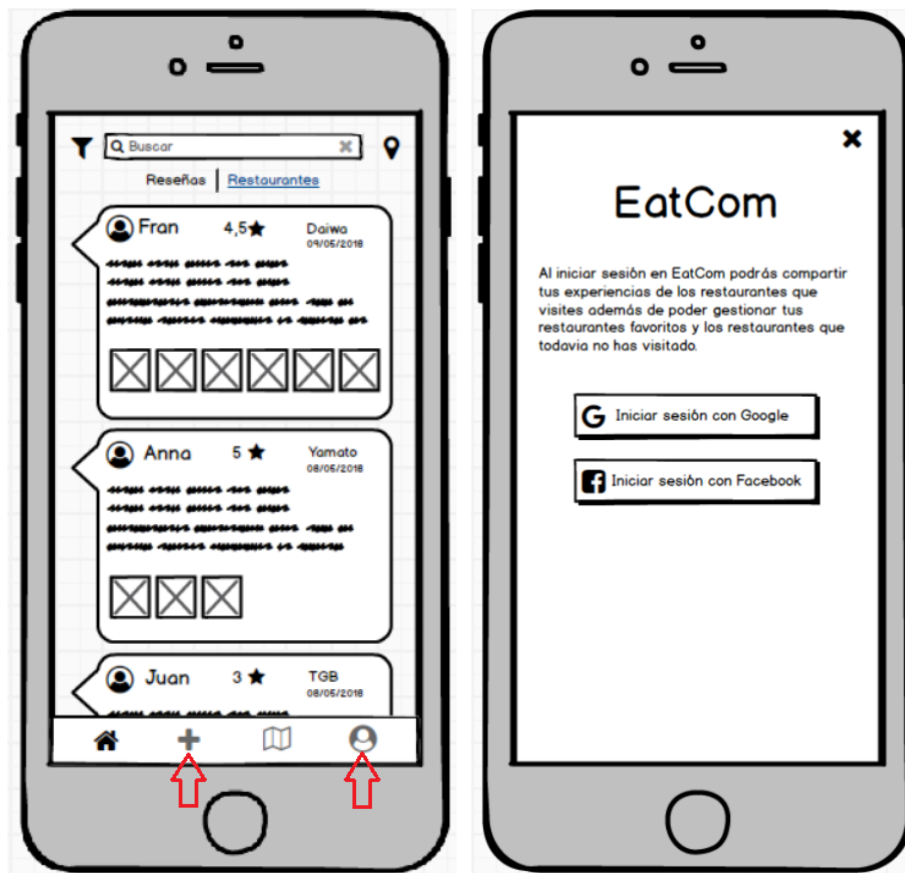


Figura 29. Identificación de usuario

Ahora que el usuario se ha identificado en la aplicación, podrá valorar los restaurantes como se indicaba en el requisito RF-06. Al pulsar en el icono de nueva reseña nos aparecerá un modal para buscar el restaurante que vamos a valorar. Una vez seleccionado el restaurante, en el siguiente paso podremos editar el restaurante si nos hemos equivocado, dar una puntuación a nuestra experiencia en el restaurante, adjuntar imágenes y describir como fue nuestra experiencia. Para poder publicar una reseña tan solo será indispensable seleccionar una valoración.



Figura 30. Escribir una reseña

Otra de las opciones que tiene el usuario cuando se ha identificado es poder ver su perfil personal, donde además de ver sus publicaciones y datos personales, podrá gestionar los restaurantes que haya guardado en la lista de favoritos o aquellos que ha guardado para ver mas tarde. En la Figura 31 quedan plasmados los requisitos RF-09 y RF-10.

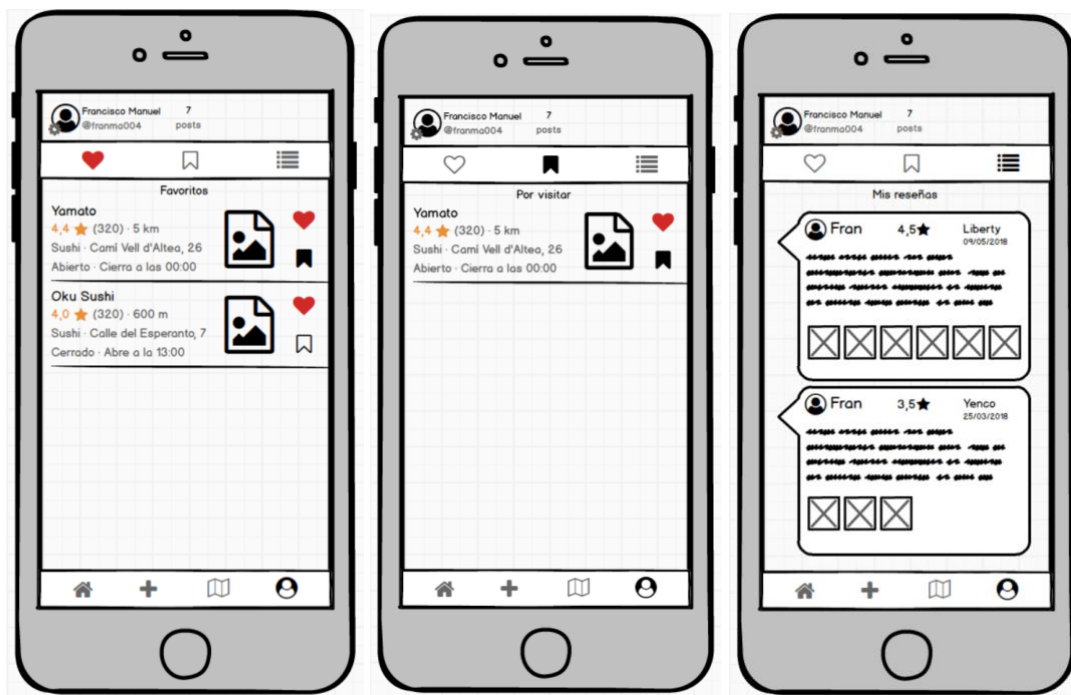


Figura 31. Perfil personal

Para abordar los requisitos RF-08 y RF-11 se ha diseñado un modal para que el usuario pueda consultar sus datos personales y modificar su alias si así lo desea o cerrar sesión si no quiere estar identificado en la aplicación.



Figura 32. Editar perfil personal

Por último, para los usuarios con perfil de administrador se ha diseñado una versión móvil en la que poder moderar las reseñas de los usuarios de forma simple y así satisfacer el requisito RF-12. Como trabajo futuro, queda pendiente hacer un backoffice web para una mejor gestión de las reseñas.

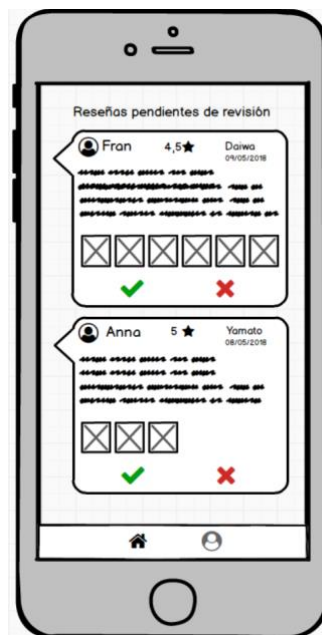


Figura 33. Vista de administrador

## 7. Implementación

Como estamos usando la metodología ágil en Cascada, la fase de implementación la hemos dividido en dos. Desarrollo de la parte de backend o el API Rest y por otro lado el frontend.

### 7.1. Entorno de desarrollo

Para el entorno de desarrollo se ha configurado una máquina virtual con un sistema operativo Ubuntu. Se ha optado por una distribución Linux por comodidad, es gratuito y ya había desarrollado algún proyecto en este entorno, por lo que la instalación y configuración de los componentes del proyecto me resultaría más ágil y rápida.

Como editor de código voy a usar Visual Studio Code con la extensión JSLint, que es una herramienta de análisis de código estático utilizada en el desarrollo de software para verificar si el código fuente de Javascript cumple con las reglas de codificación.

Para el desarrollo del proyecto, nuestro ordenador será el que haga de servidor.

### 7.2. Implementación Backend

#### 7.2.1. Instalación de dependencias

Como vamos a utilizar el MEAN stack, para el desarrollo del backend instalaremos Node.js y MongoDB. Además, Node trae consigo npm, un gestor de paquetes que se ejecuta desde la línea de comandos y nos permite instalar y desinstalar las dependencias de la aplicación.

Las dependencias básicas que hemos instalado para nuestro proyecto son:

- **Express:** Nos permite trabajar con el protocolo HTTP y se encargará de la gestión de peticiones y respuestas, cabeceras, rutas, vistas, etc.
- **Body parser:** Para poder convertir los datos que nos lleguen en las peticiones al servidor en objetos json.
- **Mongoose** es una librería para conectar Node con MongoDB que nos provee de métodos que nos permiten trabajar mejor.

Como dependencia de desarrollo vamos a instalar nodemon, una extensión que nos permitirá recargar el script de la consola de forma automática cuando hagamos cambios en nuestro

código, de esta forma no tenemos que estar constantemente reiniciando el servicio de forma manual.

Dependencias adicionales instaladas:

- Jwt-simple para la autenticación basada en token. Con esta extensión podemos codificar los datos del usuario y comprobar que es correcta.
- Connect-multiparty, librería para subir ficheros.

## 7.2.2. APIs de Google

### 7.2.2.1. Geolocation

Esta API nos devuelve la ubicación del usuario y el radio de exactitud basado en las torres WIFI que detecte el dispositivo. La latitud y longitud devueltas por esta petición serán usadas posteriormente en las llamadas a Google Places.

### 7.2.2.2. Place Search

Como hemos comentado anteriormente, no dispondremos de una base de datos con los restaurantes, sino que utilizamos Google Places para hacer consultas sobre los restaurantes y su información.

Por un lado, utilizamos Nearby Search para obtener los restaurantes dada un área seleccionada, sin necesidad de incluir texto en la búsqueda. Los parámetros que pasamos en la petición son lat, lon, radius y type. Este último es siempre *restaurant* para discriminar los resultados y solo obtener establecimientos del tipo restaurante.

La respuesta es un array de 20 restaurantes con su información y un pageToken en el que se guarda la información de la petición actual y que usamos para obtener los 20 siguientes resultados.

Para la búsqueda de restaurantes utilizamos Text Search, el cual nos devuelve los resultados en función de un string. Por ejemplo, podemos buscar “Restaurantes mexicanos” o “Sushi”, respondiéndonos con los resultados que correspondan al string enviado.

Este servicio es especialmente útil porque podemos hacer peticiones ambiguas, pudiendo incluir direcciones incompletas, las cuales el sistema sabrá identificar.

#### 7.2.2.3. Place Details

Una vez tenemos el place\_id de un restaurante de Place Search, podemos hacer una petición para obtener más detalles sobre el establecimiento. Place Details nos devuelve información más completa como la dirección completa del restaurante, número de teléfono, etc.

Todos estos servicios de Google son de pago para un uso comercial, pero dado que los datos que vamos a requerir pertenecen al grupo de Basic Data, no tendrá coste alguno. Si quisiéramos consultar más datos en un futuro como las reviews o el precio del establecimiento, el precio de 0-100000 peticiones al día es de 0.005 USD por petición, disminuyendo el precio por petición según se incrementa el volumen de peticiones.

#### 7.2.3. Controladores

Al terminar un controlador, mediante Postman podemos generar una petición para comprobar que el resultado es correcto. Este proceso se llevará a cabo para cada controlador, y las pruebas se ejecutarán cada vez que haya una modificación importante en un controlador.

### 7.3. Implementación Frontend

En la parte del frontend voy a utilizar tanto Bootstrap como Angular Material. Voy a usar Bootstrap para maquetar y usar alguna funcionalidades y Angular Material para los estilos.

Al igual que con el backend usaba nodemon para reiniciar el servicio cuando se hacían cambios en tiempo real, para el frontend voy a usar Angular-cli, que además de reiniciar el servicio, me permite generar componentes y servicios mediante comandos de forma rápida y cómoda.

Al redactar una crónica, he observado que, al subir varios ficheros de imágenes, tenía problemas para devolver los comentarios, con la información de usuario y las imágenes correspondientes, y me he dado cuenta de que fue un error de diseño del documento de Comentario. He optado por incluir un campo imágenes en el documento de Comentario, el cual será de tipo array y en el que se podrán almacenar todas las imágenes.

Para los menús de navegación se ha utilizado Angular Material para dar sensación de transición al pasar de un resultado a otro.

Para el sistema de rating, se ha usado el componente de Bootstrap para angular ngb-rating. El principal inconveniente fue una incompatibilidad de versiones con el componente, pero al actualizar la versión de angular todo fue correcto. El componen ngb-rating nos permite generar un sistema de valoración de estrellas de una forma bastante sencilla, pudiendo indicar si queremos que sea modificable, con valores decimales, el icono que se va a utilizar y la cantidad máxima.

#### 7.4. Producto final



Figura 34. Página principal – Comentarios

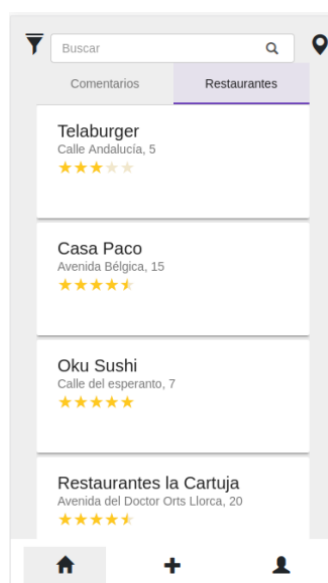
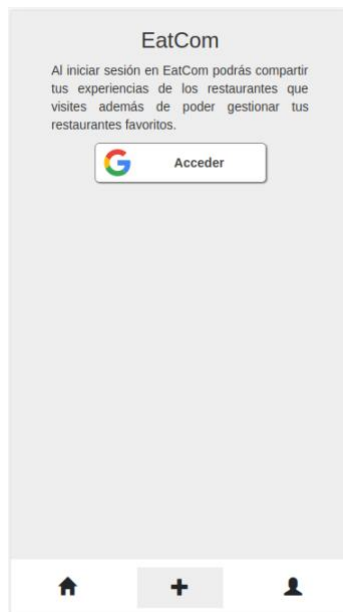


Figura 35. Página principal - Restaurantes



*Figura 36. Nueva crónica sin identificar usuario*




*Figura 37. Nueva crónica con usuario identificado*



Restaurante: Nabab

Valoración: ★★★★★

Fotos:  nabab-benidorm.jpg

Reseña:

Publicar

Home + Profile

Figura 38. Formulario para crear una crónica

F @franma004

Heart Bookmark Menu

Terra Atlántica 28/06/2018

★★★★★

Fuimos con la familia a cenar y el servicio un poco lento aunque la comida estuvo bien


X

Editar Eliminar

Oku sushi 13/06/2018

★★★★★

El mejor sushi de la zona, calidad precio es lo mejor.

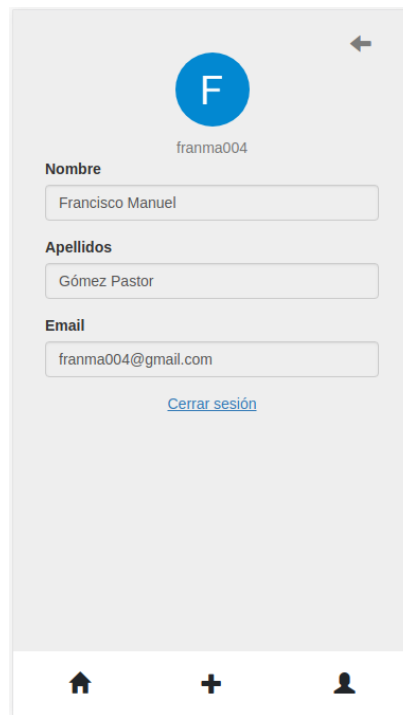


The Grill 15/05/2018

★★★★★

Home + Profile

Figura 39. Perfil de usuario



A mobile application interface for editing personal data. At the top, there is a blue circular profile picture with a white letter 'F' and the username 'franma004' below it. A back arrow is in the top right corner. The form contains three sections: 'Nombre' with the value 'Francisco Manuel', 'Apellidos' with the value 'Gómez Pastor', and 'Email' with the value 'franma004@gmail.com'. Below the email field is a blue link labeled 'Cerrar sesión'. At the bottom, there is a white navigation bar with three icons: a house, a plus sign, and a person silhouette.

←

F  
franma004

**Nombre**  
Francisco Manuel

**Apellidos**  
Gómez Pastor

**Email**  
franma004@gmail.com

[Cerrar sesión](#)

🏠 + 👤

*Figura 40. Datos personales*

## 8. Resultados

Durante el desarrollo de este proyecto, se ha diseñado e implementado un sistema que cumple con todos los requisitos funcionales. Los únicos requisitos que no se han podido llevar a cabo por razones temporales son los requisitos deseables.

La aplicación es capaz de ofrecer contenido normalizado, homogéneo y de alta precisión en la mayoría de los casos a partir de la API de Google. También permite crear crónicas sobre un restaurante en concreto, almacenando la ubicación de este y ofreciendo un listado con las crónicas de los restaurantes cercanos a la posición del usuario gracias a una funcionalidad de MongoDB. Se puede usar el buscador tanto para encontrar los restaurantes cercanos al usuario, como para encontrar comentarios que contengan las palabras clave que haya indicado el usuario.

Estoy contento con el resultado obtenido, ya que he podido aprender mucho sobre Node y Angular, que era uno de los objetivos de este TFG, aunque queda pendiente subirlo a un servidor, ya que por razones temporales no ha sido posible llevarlo a cabo.

Durante la fase de programación del frontend, el disco duro virtual que se utilizaba para arrancar la máquina quedó corrupto por un apagón mientras me encontraba programando. La parte del backend sí que se encontraba en un repositorio de Bitbucket, pero toda la parte de frontend no.

Esto causó la pérdida de mucho tiempo para poder recuperar todo el trabajo, pudiendo recuperar al final tan solo los ficheros esenciales.

Hubo que montar otro entorno de desarrollo, creando de nuevo el proyecto. Opté por instalar las últimas versiones tanto de Node como de Angular, lo que generó incompatibilidades con algunos de los métodos, los cuales hubo que adaptar a la nueva versión.

Este problema fue debido a que tenía instalada una extensión de GIT en Visual Studio Code, la cual se me olvidó configurar cuando empecé con el desarrollo de la parte del frontend.

Este proyecto no hubiera sido posible sin asignaturas como Usabilidad y accesibilidad, la cual despertó en mí el interés sobre el UI/UX. Programación Hipermedia 1 y 2 con las que sentí las bases de la programación web. Estructuración de Datos y Algoritmia, que me ha ayudado a poder estructurar mi proyecto para un desarrollo eficaz y eficiente. Y, sobre todo, Servicios Multimedia Avanzados, que fue determinante tanto para realizar este proyecto, como para decidir mi futuro laboral.

Actualmente la aplicación solo está alojada en mi máquina, por lo que para la presentación haré una demostración de uso con ella. La demostración consistirá en un recorrido guiado por la aplicación abordando las principales funcionalidades, como la creación de una crónica, búsqueda de comentarios, la autenticación con Google, etc.

## 9. Conclusiones y trabajo futuro

### 9.1. Mejoras y ampliaciones

Dada la importancia del panel de administración, queda pendiente una versión web con la capacidad de poder interactuar con los usuarios y tener más opciones administrativas.

En cuanto al sistema desarrollado, se definieron los requisitos funcionales y se clasificaron en base a su prioridad, lo que ha permitido establecer un orden en la implementación de estos y llegar a realizar todos los requisitos obligatorios.

Los requisitos deseables, se consideran mejoras que harán la aplicación más atractiva y serán tenidos en cuenta en la fase futura del proyecto.

También habría que incluir como trabajo futuro un plan de marketing, publicidad e ingresos de cara al lanzamiento de la aplicación.

### 9.2. Conclusión

El objetivo principal de este proyecto era crear un producto de principio a fin, pasando por todas las fases del proceso de desarrollo y documentando los resultados que se van obteniendo en cada una de ellas.

Abordar un proyecto desde cero me ha hecho identificar donde pueden estar los conflictos a la hora de desarrollar un proyecto, ya sea porque no se ha hecho un buen análisis de requisitos, como una mala planificación, o problemas de incompatibilidades que no se dan hasta que estás programando.

Uno de los puntos que mas me ha costado llevar a cabo ha sido la planificación, debido a que tenía que compaginar trabajo y TFG. Y al tocar tantas tecnologías nuevas, la curva de aprendizaje ha hecho que me retrasase más de lo esperado, aunque no me arrepiento en lo mas mínimo, ya que ese era uno de los objetivos y creo que lo he cumplido.

Actualmente la oferta de aplicaciones similares a la desarrollada en este proyecto es bastante alta, pero ninguna reunía todas las características presentes en este proyecto. Creo que con mas tiempo de desarrollo y añadiendo algunas funcionalidades que se quedaron en el tintero, este proyecto podría tener mucho futuro.

## Bibliografía

1. Millennials: tendencias en el uso de apps móviles, disponible en: <https://pickaso.com/2017/tendencias-uso-apps-millennials>
2. Apps o webs móviles, ¿Qué prefieren los usuarios?, disponible en: <https://www.puromarketing.com/21/19151/apps-webs-moviles-prefieren-usuarios.html>
3. Redes sociales verticales o temáticas. ¿Qué son?, disponible en: <https://www.marketingwebmadrid.es/redes-sociales-verticales-o-tematicas-que-son-ejemplos/>
4. ¿Qué es SCRUM?, disponible en: <https://proyectosagiles.org/que-es-scrum/>
5. En que consiste el modelo en cascada, disponible en: <https://openclassrooms.com/courses/gestiona-tu-proyecto-de-desarrollo/en-que-consiste-el-modelo-en-cascada>
6. ¿Qué es una base de datos relacional?, disponible en: <https://aws.amazon.com/es/relational-database/>
7. ¿Qué son las bases de datos NoSQL?, disponible en: <https://aws.amazon.com/es/nosql/>
8. Documentación Google Places, disponible en: <https://developers.google.com/places?hl=es>
9. Angular Materia, disponible en: <https://material.angular.io>